2007-06

Optimal trajectory generation for multiple asteroid rendezvous

Koeppel, David M.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/3461

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

OPTIMAL TRAJECTORY GENERATION FOR MULTIPLE ASTEROID RENDEZVOUS

by

David M. Koeppel

June 2007

Thesis Advisor: I. Michael Ross
Second Reader: Pooya Sekhavat

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2007	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Optimal Trajectory Generation for Multiple Asteroid Rendezvous			**5. FUNDING NUMBERS**
6. AUTHOR(S) David Koeppel			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

This thesis is focused on solving one component of the proposed problem in the Global Trajectory Optimization Competition released by the Jet Propulsion Laboratory in late 2006. The goal is to find an optimal spacecraft trajectory to rendezvous with an asteroid in a group of asteroids. The analysis is conducted using a MATLAB application package for dynamic optimization called DIDO. In order to verify the selection results, one-to-one transfers between Earth and several asteroids are conducted. The selection process is applied to this group of asteroids. When the initial results do not meet the expectations based on the one-to-one transfers, a more thorough search for a global minimum is necessary. The gradual cost-constrained technique is used to progress from local minima toward the global minimum. The results are checked to satisfy the constraints as well as the necessary conditions for optimality. When the results are analyzed, feasible one-to-one rendezvous trajectories are found, however a sufficient selection process is lacking. There is a great deal of work remaining on this project, including the continued development of an asteroid selection procedure.

14. SUBJECT TERMS Optimal Control, DIDO, GTOC2, Gradual Cost-Constrained Optimization			**15. NUMBER OF PAGES** 117
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

OPTIMAL TRAJECTORY GENERATION FOR MULTIPLE ASTEROID RENDEZVOUS

David M. Koeppel
Ensign, United States Navy
B.S., United States Naval Academy, 2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2007

Author: David M. Koeppel

Approved by: I. Michael Ross
Thesis Advisor

Pooya Sekhavat
Second Reader

Anthony J. Healey
Chairman, Department of Mechanical and Astronautical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis is focused on solving one component of the proposed problem in the Global Trajectory Optimization Competition released by the Jet Propulsion Laboratory in late 2006. The goal is to find an optimal spacecraft trajectory to rendezvous with an asteroid in a group of asteroids. The analysis is conducted using a MATLAB application package for dynamic optimization called DIDO. In order to verify the selection results, one-to-one transfers between Earth and several asteroids are conducted. The selection process is applied to this group of asteroids. When the initial results do not meet the expectations based on the one-to-one transfers, a more thorough search for a global minimum is necessary. The gradual cost-constrained technique is used to progress from local minima toward the global minimum. The results are checked to satisfy the constraints as well as the necessary conditions for optimality. When the results are analyzed, feasible one-to-one rendezvous trajectories are found, however a sufficient selection process is lacking. There is a great deal of work remaining on this project, including the continued development of an asteroid selection procedure.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

It would be inappropriate if I did not acknowledge, first and foremost, the assistance and encouragement that I received from Dr. Pooya Sekhavat and Professor I. M. Ross. The GNC Lab has been a great home for me over the past several months. I greatly appreciate all your patience and support while conducting this research. I would have never been able to complete this work if it weren’t for your continuous sharing of insight and assistance.

I also must thank my beautiful fiancée, Alison. Your love and support of me, even when I doubted myself, was undying and steadying. I appreciate all that you are to me and thank you for your undying patience while completing this work.

Thanks must also go out to all of the Ensigns who have supported me in so many ways during our year together in Monterey. My time here would not have been the same if it weren’t for all of you.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

On 5 October 2006, the Outer Planets Mission Analysis Group of the Jet Propulsion Laboratory (JPL) released the second problem in the Global Trajectory Optimization Competition (GTOC2). The problem specified is a low-thrust trajectory design for a multiple asteroid rendezvous. The goal is to maximize the ratio of the final mass of the spacecraft to flight time. There are four clusters of asteroids to be met in sequence. The spacecraft is to rendezvous with one asteroid in each of four asteroid clusters. The order in which the spacecraft visits the clusters is inconsequential. This problem was chosen by JPL based on four criteria: 1) global optimization consists of several local optima over a large launch window, 2) unusual objective function or constraints, 3) the problem can be solved in a reasonable amount of time, and 4) solutions to the problem can be easily verified.[1] The goal of this research is to solve part of this problem, the selection of an asteroid in a group of asteroids which provides the optimal trajectory for rendezvous.

B. THE PROBLEM

This thesis focuses on the selection of an individual asteroid in a cluster with which to rendezvous, such that the cost is minimized; this work, therefore, is only addressing one component of the GTOC2 problem. The problem is simplified by limiting the forces acting on the spacecraft to only the gravitational force of the sun and the thrust provided by the spacecraft's engine. As such, there can be no gravity assists used in the solution of the problem. The propulsion for the spacecraft is provided by a thruster which can be turned on

[1] Anastassios E. Petropoulos, "Problem Description for the 2nd Global Trajectory Optimization Competition," *Outer Planets Mission Analysis Group, Jet Propulsion Laboratory*, 6 November 2006.

and off at will. The direction of the propulsion is unconstrained. The specifications for the engine are provided in Table 1.[2]

Table 1: Spacecraft engine specifications

Constant Specific Impulse	4000 s
Maximum Thrust Level	0.1 N

There is a twenty year window for the launch date, which must fall between 2015 and 2035. The spacecraft is to launch from the Earth with a hyperbolic escape velocity of 3.5 km/s. The direction of this velocity is unconstrained.

According to the problem statement, the Earth and asteroids can be assumed to follow Keplerian (conic) orbits about the Sun, eliminating perturbations and thus further simplifying the problem. The Keplerian orbital elements for the Earth are provided in Table 2.[3]

Table 2: Keplerian orbit elements of the Earth, J2000 heliocentric ecliptic reference frame

Semimajor axis, a (AU)	0.999988049532578
Eccentricity, e	0.01671681163160
Inclination , i (deg)	0.0008854353079654
Longitude of ascending node, Ω (deg)	175.40647696473
Argument of perigee, ω (deg)	287.61577546182
Mean anomaly at epoch, M_0 (deg)	257.60683707535
Epoch, t_0 (MJD)	54000

[2] Anastassios E. Petropoulos, "Problem Description for the 2nd Global Trajectory Optimization Competition."

[3] Ibid.

The constants and conversions which are to be used in solving this problem are provided in Table 3.[4]

Table 3: Constants and conversions

Gravitational parameter of the Sun, μ (km^3/s^2)	$1.32712440018 \times 10^{11}$
Astronomical unit (AU), (km)	$1.49597870681 \times 10^{8}$
Standard acceleration due to gravity, g (m/s^2)	9.80665
Day, (s)	86400
Year, (days)	365.25
00:00 01 January 2015, (MJD)	57023.5
24:00 31 December 2035, (MJD)	64693.5

The data for the asteroids are provided in an ASCII text file, containing the Keplerian orbital elements (see Appendix A). The following data are specified for each asteroid: 1) a unique ID number of the asteroid; 2) semimajor axis in AU; 3) eccentricity; 4) inclination in degrees; 5) longitude of the ascending node in degrees; 6) argument of periapsis in degrees; 7) mean anomaly in degrees; 8) epoch, in MJD, at which mean anomaly is given; and 9) the group number of the asteroid. Just as they are for Earth, the orbital elements of the asteroids are given in the J2000 heliocentric ecliptic frame.[5] From all of these provided values, constants and constraints, the dynamics of the problem can be developed. This will be shown in Chapter II.

C. HISTORICAL BACKGROUND

The motion of celestial bodies has been studied since the onset of civilization. In many ways, the level of development of a civilization can be determined by their knowledge of the workings of the heavens. The ability of

[4] Anastassios E. Petropoulos, "Problem Description for the 2nd Global Trajectory Optimization Competition."

humans to predict the motion of objects in space has become quite sophisticated through time, taking into account minuscule forces which affect the objects in minute, but distinguishable ways. For this research, the lack of perturbations to the orbits permits the simple use of the equations generated by Johannes Kepler and Isaac Newton to predict the motion of the bodies. These two mathematicians generated equations that could predict the position of celestial bodies based on careful observations. Kepler used the data collected throughout Tycho Brahe's life to come up with his three laws; 1) the orbits of the planets are ellipses with the Sun at one focus, 2) the line joining a planet to the Sun sweeps out equal areas in equal times and 3) the square of the orbital period is directly proportional to the cube of the mean distance between the Sun and the planet.[6] A century later, Newton would use Galileo's studies of falling objects to create his universal law of gravitation. This law states that the force of gravity between two bodies is directly proportional to the product of their masses divided by the square of the distance separating them. These simple ideas, developed in the 16th and 17th centuries, will be used to develop the dynamics of the problem at hand.

The desire for optimized solutions traces its roots back to the 17th century when scientists and mathematicians such as Galileo Galilei and Pierre de Fermat posed problems which necessitated optimization. In order to solve these problems, Isaac Newton invented calculus of variations.[7] Johannes Bernoulli was also instrumental in the initial surge for optimization as he used a discrete-step method to solve the brachistochrone problem in 1697. Many other great minds, such as Leonard Euler, Jean Louis Lagrange, Adrien-Marie Legendre (who first introduced multiple variable variations), Karl Gustav Jacob Jacobi and

---

5 Anastassios E. Petropoulos, "Problem Description for the 2nd Global Trajectory Optimization Competition."

6 Jerry Jon Sellers, *Understanding Space: An Introduction to Astronautics* (Boston: McGraw Hill Custom Publishing, 2004), 36-39.

7 Arthur E. Bryson, *Dynamic Optimization* (Menlo Park, CA: Addison-Wesley Longman, Inc., 1999), 403-405.

William Rowan Hamilton, made contributions to the field in the 18th and 19th centuries.[8] Numerous others have continued the rapid development and advancement of optimal control theory up to present day including Lev Pontryagin and Richard Bellman.

D. PSEUDOSPECTRAL METHOD

One of the most modern control theories is based on pseudospectral approximation. The pseudospectral (PS) approach can be implemented to solve dynamic optimization problems. It is used to solve the optimal control problem in this work. Here, we take a closer look at the theory behind it.

Several pseudospectral methods can be used to solve optimal control problems. In this work, the Legendre Pseudospectral Method is utilized to numerically solve the dynamic optimal control problem. The advantage of the Legendre PS method is the ability to vary the spacing between data points.[9] By placing the nodes at the roots of the Legendre polynomials, the integration of any polynomial of degree 2N-1 can be exactly evaluated. The Legendre PS method places the nodes at the zeros of the derivatives of the N-1 degree Legendre polynomials $P_{N-1}(t)$. The values at the endpoints are reserved for i=1 and i=N. The derivation of matrix D^N necessary for implementation of the Legendre PS method is now possible through differentiation of the interpolation polynomial.[10]

$$x^N(t_i) = \sum_{j=1}^{N} x(t_j)\phi_j(t_i) \tag{1.1}$$

$$\dot{x}^N(t_i) = \sum_{j=1}^{N} x(t_j)\dot{\phi}_j(t_i) = \sum_{j=1}^{N} D_{ij}x_j \tag{1.2}$$

$$D_{ij} = \dot{\phi}_j(t_i) \tag{1.3}$$

---

8 Arthur E. Bryson, *Dynamic Optimization*, 403-405.

9 I. M. Ross and F. Fahroo, “Pseudospectral Methods for Optimal Motion Planning of Differentially Flat Systems,” (paper presented at IEEE Conference on Transactions on Automatic Control, Las Vegas, NV, Dec. 2002).

10 I. M. Ross and F. Fahroo, “Legendre Pseudospectral Approximations of Optimal Control Problems,” Lecture Notes in Control and Information Sciences (2003).

The next step requires the interpolating polynomial to be in terms of the Legendre polynomial. The intermediate function is defined as:

$$z(t) = \prod_{i=1}^{N} t - t_i \tag{1.4}$$

The derivative evaluated at a root, j, results in the following relationship.

$$\dot{z}(t_j) = \left(t_j - t_1\right)\ldots\left(t_j - t_{j-1}\right)\left(t_j - t_{j+1}\right)\ldots\left(t_j - t_N\right) = \prod_{\substack{i=1 \\ i \neq j}}^{N} t_j - t_i \tag{1.5}$$

Now, the Lagrange interpolating function is expressed in terms of the intermediate function (Eq. (1.4)).[11]

$$\phi_j(t) = \prod_{\substack{i=1 \\ i \neq j}}^{N} \frac{\left(t - t_i\right)}{\left(t_j - t_i\right)} = \frac{z(t)}{(t - t_j)\dot{z}(t_j)} \tag{1.6}$$

The locations of the nodes $\{t_2, \ldots, t_{N-1}\}$ are chosen to be the roots of the derivative of the Legendre polynomials $\left(P'_{N-1}(t)\right)$. It can be seen that the intermediate function has the same roots, with additional roots at the endpoints (1 and -1).[12]

$$z(t) = (t - t_1)P'_{N-1}(t)(t - t_N) = (t+1)P'_{N-1}(t)(t-1) = (t^2 - 1)P'_{N-1}(t) \tag{1.7}$$

The Legendre differential equation is shown below.

$$\frac{d}{dt}\left[(t^2 - 1)P'_{N-1}(t)\right] = N(N-1)P_{N-1}(t) \tag{1.8}$$

Applying the Legendre differential equation to the intermediate equation yields the following equation.

$$\dot{z}(t) = N(N-1)P_{N-1}(t) \tag{1.9}$$

Combining Eqs. (1.6), (1.7), and (1.9) results in the following.

$$\phi_j(t) = \frac{(t^2 - 1)P'_{N-1}(t)}{(t - t_j)N(N-1)P_{N-1}(t_j)} \tag{1.10}$$

[11] I. M. Ross and F. Fahroo, "Legendre Pseudospectral Approximations of Optimal Control Problems."

[12] I. M. Ross and F. Fahroo, "Pseudospectral Methods for Optimal Motion Planning of Differentially Flat Systems."

Evaluating the derivative of (1.10) at $t = t_i$ will results in the following equation.[13]

$$\dot{\phi}_j(t_i) = \frac{1}{N(N-1)P_{N-1}(t_j)}\left(\frac{2t_i P'_{N-1}(t_i)}{t_i - t_j} + \frac{(t_i^2-1)P''_{N-1}(t_i)}{t_i - t_j} - \frac{(t_i^2-1)P'_{N-1}(t_i)}{(t_i - t_j)^2}\right) \quad (1.11)$$

Now, the Legendre differential equation (Eq. (1.8)) is simplified,

$$N(N-1)P_{N-1}(t) = \frac{d}{dt}\left[(t^2-1)P'_{N-1}(t)\right] = 2tP'_{N-1}(t) + (t^2-1)P''_{N-1}(t) \qquad (1.12)$$

And combined with Eq. (1.11) as follows.[14]

$$\dot{\phi}_j(t_i) = \frac{1}{N(N-1)P_{N-1}(t_j)}\left(\frac{N(N-1)P_{N-1}(t_i)}{t_i - t_j} - \frac{(t_i^2-1)P'_{N-1}(t_i)}{(t_i - t_j)^2}\right) \qquad (1.13)$$

Since $(t_i^2-1)P'_{N-1}(t_i) = 0$, the differentiation matrix, D_{ij}, can now be produced as:

$$D_{ij} = \dot{\phi}_j(t_i) = \frac{P_{N-1}(t_i)}{(t_i - t_j)P_{N-1}(t_j)} \qquad (1.14)$$

The values which fill the differentiation matrix require further manipulation and analysis. For example, L'Hôpital's rule must be applied to Eq. (1.13) so that the $t_i - t_j$ terms can be eliminated from the denominator. From Eq. (1.14), the components of D_{ij} that form the N x N differentiation matrix D^N, are:[15]

$$D_{ij} = \begin{cases} \dfrac{P_N(t_i)}{P_N(t_j)} \cdot \dfrac{1}{t_i - t_j}, & i \neq j \\ -\dfrac{N(N-1)}{4}, & i = j = 1 \\ \dfrac{N(N-1)}{4}, & i = j = n \\ 0 & \text{otherwise} \end{cases} \qquad (1.15)$$

This completes the brief explanation of the numerical technique used in the Legendre Pseudospectral Method. A complete understanding of this theory

[13] I. M. Ross and F. Fahroo, "Legendre Pseudospectral Approximations of Optimal Control Problems."

[14] Ibid.

[15] I. M. Ross and F. Fahroo, "Pseudospectral Methods for Optimal Motion Planning of Differentially Flat Systems."

is not necessary for this work, as these calculations will be completed by DIDO, a MATLAB based application package authored at the Naval Postgraduate School. This will be explained in greater depth in the subsequent chapter.

E. CONCLUSION

The problem to be solved in this research is straightforward. The equations of motion will be derived based on simple two body orbital mechanics. There are several assumptions that are made which simplify the solution process (e.g. no gravity assists). The solution, however, is in no way simple to obtain. Optimal control theory will be applied to the dynamics governing the motion of the bodies such that the cost of the mission can be minimized. This problem is unique in that the minimized cost is not just the mass of the propellant or the time to travel the trajectory. Instead, the goal is to maximize the ratio of final mass and travel time, implementing both mass and time optimization.

II. THE PROBLEM FORMULATION

A. INTRODUCTION

When JPL posed GTOC2, it was intended to be a problem that could be solved in a reasonable period of time by a team of experienced researchers. Therefore, the astrodynamics were simplified to a two body problem with only the gravity of the sun acting on the bodies.

B. GOVERNING DYNAMICS

The equations of motion of the problem stem from Newton's laws. The only force acting on the Earth and the asteroids is the gravitational pull of the sun. Therefore, by combining Newton's second law, $\Sigma F = m \cdot a$, with his law of gravitation, $F = \frac{-G \cdot M \cdot m}{r^2} = \frac{-\mu \cdot m}{r^2}$, the equations of motion of the Earth and asteroids can be determined:[16]

$$\begin{aligned} \ddot{x} + \mu \frac{x}{r^3} &= 0 \\ \ddot{y} + \mu \frac{y}{r^3} &= 0 \\ \ddot{z} + \mu \frac{z}{r^3} &= 0 \end{aligned} \tag{2.1}$$

These are the two-body equations of motion in Cartesian coordinates (x, y, z), where $r = \sqrt{x^2 + y^2 + z^2}$. The gravitational constant, μ, in this case is that of the Sun, the value of which can be found in Table 3.

[16] Daryl G. Boden, "Introduction to Astrodynamics," in *Spacecraft Mission Analysis and Design,* ed. Wiley J Larson and James R. Wertz (El Segundo, CA: Microcosm Press, 1999), 132.

The solution to the generic two body equation ($\ddot{r} + \frac{\mu}{r^3} = 0$) is known to be the polar equation of a conic section.[17]

$$r = \frac{a \cdot (1 - e^2)}{\left(1 + e \cdot \cos(\nu)\right)} \tag{2.2}$$

The velocity of an object in an elliptical orbit is determined by the following equation.[18]

$$v = \sqrt{\frac{2\mu}{r} - \frac{\mu}{a}} \tag{2.3}$$

At this point, the complexity of converting between the elliptical coordinates (in which the orbital elements are presented in) and the Cartesian coordinates (in which the equation of motion are derived), begins to become apparent. In order to convert between these reference frames, the following relationships are used.[19]

$$\begin{aligned}
x &= r\left[\cos(\theta + \omega) \cdot \cos(\Omega) - \sin(\theta + \omega) \cdot \cos(i) \cdot \sin(\Omega)\right] \\
y &= r\left[\cos(\theta + \omega) \cdot \sin(\Omega) + \sin(\theta + \omega) \cdot \cos(i) \cdot \cos(\Omega)\right] \\
z &= r\left[\sin(\theta + \omega) \cdot \sin(i)\right] \\
v_x &= v\left[-\sin(\theta + \omega - \gamma) \cdot \cos(\Omega) - \cos(\theta + \omega - \gamma) \cdot \cos(i) \cdot \sin(\Omega)\right] \\
v_y &= v\left[-\sin(\theta + \omega - \gamma) \cdot \sin(\Omega) + \cos(\theta + \omega - \gamma) \cdot \cos(i) \cdot \cos(\Omega)\right] \\
v_z &= v\left[\cos(\theta + \omega - \gamma) \cdot \sin(i)\right]
\end{aligned} \tag{2.4}$$

These equations requite the flight path angle (γ). This angle can be found from the following equation:

$$\tan(\gamma) = \frac{e \cdot \sin(\theta)}{1 + e \cdot \cos(\theta)} \tag{2.5}$$

which requires the value of the true anomaly (θ). True anomaly can be found from:

---

[17] Jerry Sellers, *Understanding Space,* 136-137.

[18] Ibid.

[19] Anastassios E. Petropoulos, "Problem Description for the 2nd Global Trajectory Optimization Competition."

$$\tan\left(\frac{E}{2}\right) = \sqrt{\frac{1-e}{1+e}}\tan\left(\frac{\theta}{2}\right) \tag{2.6}$$

This equation contains the eccentric anomaly, E, whose value is found by using Kepler's Equation, shown below:

$$M = E - e \cdot \sin(E) \tag{2.7}$$

Kepler's equation is a nonlinear equation that must be solved in an iterative process to calculate eccentric anomaly.[20] It also requires the knowledge of mean anomaly, M, calculated from the following equation:

$$M - M_0 = \sqrt{\frac{\mu}{a^3}}\left(t - t_0\right) \tag{2.8}$$

Where M_0 is the initial mean anomaly of the object, t_0 is the initial time when the object is at M_0, a is the semimajor axis of the orbit of the body and μ is that of the Sun. Eqs. (2.2) through (2.8) yield the Cartesian coordinates of the bodies in motion.

The only difference between the equations of motion for the spacecraft and those of the celestial bodies is the addition of the spacecraft thrust force, $\bar{T}$. This force can be projected into its Cartesian components.

$$\bar{T} = \sqrt{T_x{}^2 + T_y{}^2 + T_z{}^2} \leq 0.1\,N \tag{2.9}$$

The spacecraft equations of motion include this additional force as follows:[21]

$$\begin{aligned} \ddot{x} + \mu\frac{x}{r^3} &= \frac{T_x}{m} \\ \ddot{y} + \mu\frac{y}{r^3} &= \frac{T_y}{m} \\ \ddot{z} + \mu\frac{z}{r^3} &= \frac{T_z}{m} \end{aligned} \tag{2.10}$$

The mass of the spacecraft will decrease as fuel is used to create thrust. Therefore, the last equation of motion is that of the mass flow rate of propellant:

---

[20] Anastassios E. Petropoulos, "Problem Description for the 2nd Global Trajectory Optimization Competition."

[21] Ibid.

$$\dot{m} = \frac{-T}{I_{sp} \cdot g} \tag{2.11}$$

Having developed the relationships which govern the motion of the Earth, asteroids and the spacecraft, the dynamics of the problem is established. These dynamic equations will constrain the solution of the optimal control problem and will be included in the optimal control problem definition.

C. OPTIMAL CONTROL PROBLEM

In this section the optimal control problem is formulated. We will establish the notation and methodology that will be used to solve the problem. The control function, $\underline{u}^*(t)$, and the state function which results from the controls, $\underline{x}^*(t)$, must be established such that the Bolza cost functional is minimized. The Bolza cost functional is shown below.[22]

$$J(\underline{x}(\cdot),\underline{u}(\cdot),t_f) = E(\underline{x}(t_f),t_f) + \int_{t_0}^{t_f} F(\underline{x}(t),\underline{u}(t),t)dt \tag{2.12}$$

In the Bolza cost functional, $\underline{x} \in \mathbb{R}^n$ and $\underline{u} \in \mathbb{R}^m$ are constrained by the differential constraint $\dot{\underline{x}} = f\left(\underline{x}(t),\underline{u}(t)\right)$ where $t \in \left[t_0,t_f\right]$. The boundary conditions can then be specified as:[23]

$$\begin{aligned} &\underline{e}_0(\underline{x}(t_0),t_0) = 0 \quad e_0 \in \mathbb{R}^p \text{ and } p \le n+1 \\ &\underline{e}_f(\underline{x}(t_f),t_f) = 0 \quad e_f \in \mathbb{R}^q \text{ and } q \le n+1 \end{aligned} \tag{2.13}$$

For this problem, the Bolza cost functional to be maximized is the ratio of final mass to flight time.

$$J_1 = \frac{m_f}{t_f - t_0} \tag{2.14}$$

For this research, the cost function will be the inverse of this value. This will allow for minimization of the cost function. Therefore, the Bolza cost functional which will be used for this analysis is:

[22] Arthur E. Bryson, *Dynamic Optimization*.

[23] Ibid.

$$J = \frac{1}{J_1} = \frac{t_f - t_0}{m_f} \tag{2.15}$$

Based on the dynamics derived in Section B of this chapter, the state and control variables are:

$$\begin{gathered} \underline{x} \in \mathbb{R}^7 \quad \{x, y, z, m, v_x, v_y, v_z\} \\ \underline{u} \in \mathbb{R}^5 \quad \left\{ (T_x, T_y, T_z, T, r) :\ T_x^{\,2} + T_y^{\,2} + T_z^{\,2} = T^2;\ \mathrm{x}^2 + y^2 + z^2 = r^2 \right\} \end{gathered} \tag{2.16}$$

With knowledge of the governing dynamics of the problem (developed in Section B), and the cost function (Eq. (2.15)), the problem formulation for the spacecraft rendezvous is developed as follows:

$$\begin{aligned} &\text{Minimize} && \mathrm{J}[\mathrm{x}(\cdot), \mathrm{u}(\cdot), \mathrm{t_f}] = \frac{t_f - t_0}{m_f} \\ &\text{Subject to} && \dot{x} = v_x \\ &&& \dot{y} = v_y \\ &&& \dot{z} = v_z \\ &&& \dot{m} = \frac{-T}{g \cdot I_{sp}} \\ &&& \dot{v}_x = \frac{T_x}{m} - \mu \frac{x}{r^3} \\ &&& \dot{v}_y = \frac{T_y}{m} - \mu \frac{y}{r^3} \\ &&& \dot{v}_z = \frac{T_z}{m} - \mu \frac{z}{r^3} \end{aligned}$$

$$\begin{gathered} \left(x_0, \mathrm{y}_0, \mathrm{z}_0, m_0, \mathrm{v}_{\mathrm{x}_o}, \mathrm{v}_{\mathrm{y}_o}, \mathrm{v}_{\mathrm{z}_o} \right) = \left(x_{earth_o}, y_{earth_o}, z_{earth_o}, 1500\ \mathrm{kg}, v_{xearth_o}, v_{yearth_o}, v_{zearth_o} \right) \\ \left(x_f, \mathrm{y}_f, z_f, \mathrm{v}_{\mathrm{x}_f}, \mathrm{v}_{\mathrm{y}_f}, \mathrm{v}_{\mathrm{z}_f} \right) = \left(x_{asteroid_f}, y_{asteroid_f}, z_{asteroid_f}, v_{xasteroid_f}, v_{yasteroid_f}, v_{zasteroid_f} \right) \end{gathered} \tag{2.17}$$

This is the problem that must be formulated and fed into DIDO to generate an optimal trajectory.

D. SOLUTION METHOD

Optimal control solution methods have been divided into two categories: indirect and direct methods. In indirect methods, Pontryagin's Minimum Principle

is used to derive the necessary conditions and to obtain the optimal trajectory. Indirect methods are known to require rigorous efforts. Direct methods discretize the optimal control problem into a parameter optimization problem. The result is a nonlinear programming problem which can be solved based on known nonlinear programming solution methods. The MATLAB-based software package, DIDO, will be used to solve this problem. DIDO uses a direct method to solve optimal control problems.

DIDO is a reusable software package which utilizes the Legendre Pseudospectral method to numerically solve partial differential equations. The Legendre Pseudospectral method is also commonly used in fluid mechanics. The uniqueness of the method comes from its use of global orthogonal polynomials as trial functions.[24] It requires that the time domain be discretized into a set of Legendre-Gauss-Lobatto (LGL) points which will henceforth be referred to as nodes. The values of the state and control variables at the nodes are the unknowns while the Lagrange polynomials act as the trial functions.[25] Based on the Lagrange polynomials, the nonlinear state equations are transformed into nonlinear algebraic functions. The result is a nonlinear programming problem which can be solved using a sparse numerical optimizer. The relationship between the Lagrange variables and the costate variables allows the costates to be determined at the nodes based on the Covector Mapping Theorem.[26] The Covector Mapping Theorem suggests that given any feasible solution to a problem, there exists a feasible solution to the augmented

---

[24] I. M. Ross and F. Fahroo, "User's Manual for DIDO 2001: A MATLAB Application Package for Dynamic Optimization" Tec. Rep. NPS Technical Report AA-01-003, Department of Aeronautics and Astronautics, (Naval Postgraduate School, Monterey, CA, 2001).

[25] Ibid.

[26] Qi Gong, I. M. Ross, Wei Kang and F. Fahroo, "On the Pseudospectral Covector Mapping Theorem for Nonlinear Optimal Control" (paper presented at the 45th IEEE Conference on Decision & Control, San Diego, CA, Dec 13-15, 2006).

Karush-Kuhn-Tucker conditions, within a feasibility tolerance.[27] The Karush-Kuhn-Tucker conditions state the following.[28]

$$\mu_i \quad \begin{cases} \leq 0 & \\ = 0 & \text{if} \\ \geq 0 & \\ \text{unrestricted} & \end{cases} \quad \begin{aligned} & h_i(u,t) = h_i^L(t) \\ & h_i^L(t) < h_i(u,t) < h_i^U(t) \\ & h_i(u,t) = h_i^U(t) \\ & h_i^L(t) = h_i^U(t) \end{aligned} \tag{2.18}$$

where μ_i is the covector associated with path constraints and h_i^L and h_i^U are the lower and upper bounds of the path constraint function (h_i). Knowing the values of the costates allows for the validation of the optimality of the provided solution, as explained in the next section.[29] More information on Pseudospectral methods can be found elsewhere.

E. NECESSARY CONDITIONS FOR OPTIMALITY

In this section, the the necessary conditions for optimality will be developed. These conditions are used to verify the optimality of a solution. The endpoint Lagrangian is known to be:

$$\bar{E} = E + \nu^T \underline{e} \tag{2.19}$$

Where E is the endpoint cost function, ν is the covector associated with endpoint constraints and $\underline{e}$ is the endpoint constraint function. The endpoint constraints present in the problem are the initial position, velocity and mass of the Earth and the final position and velocity of the asteroid.

$$\underline{e} = \left[\bar{x}_0, \bar{v}_0, m_0, \bar{x}_f, \bar{v}_f, M_{Earth_0}, M_{Ast_f} \right] \tag{2.20}$$

---

[27] Qi Gong, I. M. Ross, Wei Kang and F. Fahroo, “On the Pseudospectral Covector Mapping Theorem for Nonlinear Optimal Control.”

[28] Qi Gong, I. M. Ross and Wei Kang, “A Pseudospectral Method for the Optimal Control of Constrained Feedback Linearizable Systems” (paper presented at the 2005 IEEE Conference on Control Applications, Toronto, Canada, August 28-31, 2005).

[29] Ibid.

The Hamiltonian is defined by the following relationship:

$$H(\underline{\lambda},\underline{x},\underline{u},t) = F(\underline{x},\underline{u},t) + \lambda^T f(\underline{x},\underline{u},t) \tag{2.21}$$

where $t \to \lambda \in \mathbb{R}^{N_x}$ is a Lagrange multiplier function, also known as a costate.[30] The costate must satisfy the differential equation known as the adjoint equation:

$$\dot{\lambda} = -\frac{\partial H}{\partial x} \tag{2.22}$$

The state dynamics joined with the adjoint equation form a Hamiltonian system. For a control function $u(\cdot)$ to be optimal, it is necessary that u globally minimize the Hamiltonian for any $t \in [t_0, t_f]$.[31] Ultimately, a candidate solution for optimal control results from the minimization of the Hamiltonian with respect to u if λ and x are held constant. This is known as the Hamiltonian Minimization Condition. A candidate control function, $u^*(\cdot)$, can be generated at each t. This control solution is known as the extremal control.

$$(\lambda, x, t) \mapsto u^* \tag{2.23}$$

Pontryagin's Minimum principle develops the necessary conditions for optimality of the control u* based on the following equations.[32]

$$\text{Hamiltonian Minimization} \to H(\lambda^*, x^*, u^*, t) \le H(\lambda^*, x^*, u, t) \tag{2.24}$$

$$\text{Adjoint equations} \to \dot{\lambda}^* = \frac{-\partial H}{\partial x} \tag{2.25}$$

$$\text{Initial transversality} \to \lambda(t_0) = -\nu^T \frac{\partial e_0}{\partial x_0} \tag{2.26}$$

$$\text{Terminal transversality} \to \lambda(t_f) = \frac{\partial E}{\partial x_f} + \nu^T \frac{\partial e_f}{\partial x_f} \tag{2.27}$$

$$\text{Hamiltonian Value} \to H[t_f] + \frac{\partial E}{\partial t} + \nu^T \frac{\partial e_f}{\partial t} = 0 \tag{2.28}$$

Based on the problem formulation in Eq. (2.17) and $F(\underline{x},\underline{u},t) = 0$ the Hamiltonian then becomes:

---

30 I. M. Ross, *Control and Optimization: An introduction to Principles and Applications, Electronic Edition*, Naval Postgraduates School, Monterey, CA, December, 2005.

31 Ibid.

32 Arthur E. Bryson, *Dynamic Optimization*.

$$H(\lambda,x,u,t) = \lambda_x v_x + \lambda_y v_y + \lambda_z v_z + \lambda_m \frac{-T}{g \cdot I_{sp}} + \lambda_{v_x}\left(\frac{T_x}{m} - \mu\frac{x}{r^3}\right) + \lambda_{v_y}\left(\frac{T_y}{m} - \mu\frac{y}{r^3}\right) + \lambda_{v_z}\left(\frac{T_z}{m} - \mu\frac{z}{r^3}\right) \tag{2.29}$$

Although it does not directly appear in Eq. (2.29), all of the terms are time dependant. This is a result of the time dependence of mean anomaly (Eq. (2.8)) as developed as a part of the equations of motion derived in Section B.

$$\frac{\partial H}{\partial t} \neq 0 \tag{2.30}$$

The end-point Lagrangian can be defined, allowing the final value of the Hamiltonian to be determined. This information is useful when determining the validity of a solution produced by DIDO.

$$H\left[t_f\right] + \frac{\partial \bar{E}}{\partial t_f} = 0 \tag{2.31}$$

Since the endpoint conditions in this problem are also time dependant through the propagation of mean anomaly, $\bar{E}$ is dependent on time:

$$\frac{\partial \bar{E}}{\partial t_f} = \frac{\partial E}{\partial t_f} + \frac{\partial(\nu^T \underline{e})}{\partial t_f} = \frac{1}{m_f} + \frac{\partial(\nu^T \underline{e})}{\partial t_f} \tag{2.32}$$

$$\therefore H\left[t_f\right] = -\frac{1}{m_f} - \frac{\partial(\nu^T \underline{e})}{\partial t_f} \tag{2.33}$$

The adjoint equations can also be developed from the Hamiltonian (see Eq. (2.25)). The adjoint equation defined earlier is applied to the Hamiltonian Lagrangian, $\bar{H}$, defined as:

$$\bar{H}\left(\bar{\mu},\bar{\lambda},\bar{x},\bar{u},t\right) := H\left(\bar{\lambda},\bar{x},\bar{u},t\right) + \bar{\mu}^T \cdot \bar{h}\left(\bar{u}\right) \tag{2.34}$$

where:

$$\bar{h}(\bar{u}) = \begin{bmatrix} h_1(\bar{u}) \\ h_2(\bar{u}) \end{bmatrix} = \begin{bmatrix} T_x^2 + T_y^2 + T_z^2 - \bar{T}^2 = 0 \\ x^2 + y^2 + z^2 - r^2 = 0 \end{bmatrix} \tag{2.35}$$

The adjoint equations are therefore:

$$\dot{\bar{\lambda}}^* = \frac{-\partial \bar{H}}{\partial \underline{x}}$$

$$-\dot{\lambda}_x = \frac{\partial \bar{H}}{\partial x} = -\lambda_{v_x} \cdot \frac{\mu}{r^3} - 2\mu_2 x$$

$$-\dot{\lambda}_y = \frac{\partial \bar{H}}{\partial y} = -\lambda_{v_y} \cdot \frac{\mu}{r^3} - 2\mu_2 y$$

$$-\dot{\lambda}_z = \frac{\partial \bar{H}}{\partial z} = -\lambda_{v_z} \cdot \frac{\mu}{r^3} - 2\mu_2 z$$

$$-\dot{\lambda}_m = \frac{\partial \bar{H}}{\partial m} = \lambda_{v_x} \cdot \frac{T_x}{m^2} + \lambda_{v_y} \cdot \frac{T_y}{m^2} + \lambda_{v_z} \cdot \frac{T_z}{m^2}$$

$$-\dot{\lambda}_{v_x} = \frac{\partial \bar{H}}{\partial v_x} = -\lambda_x$$

$$-\dot{\lambda}_{v_y} = \frac{\partial \bar{H}}{\partial v_y} = -\lambda_y$$

$$-\dot{\lambda}_{v_z} = \frac{\partial \bar{H}}{\partial v_z} = -\lambda_z \tag{2.36}$$

Using Eq. (2.36), the dual variable outputs can be analyzed and used for further verification of the solution.

Another option for analytically evaluating the optimality of a solution is through the analysis of the transversality conditions (Eqs. (2.26) and (2.27)). The endpoint constraint function of this problem is affected by changes in the states.

$$\lambda(t_0) = -\nu^T \frac{\partial \underline{e}_0}{\partial \underline{x}_0} = -\nu_1 \frac{\partial \underline{e}_1}{\partial \underline{x}_0} - \nu_2 \frac{\partial \underline{e}_2}{\partial \underline{x}_0} - \nu_6 \frac{\partial \underline{e}_6}{\partial \underline{x}_0} \tag{2.37}$$

$$\lambda(t_f) = \frac{\partial \bar{E}}{\partial \underline{x}_f} + \nu^T \frac{\partial \underline{e}_f}{\partial \underline{x}_f} = 0 + \nu_4 \frac{\partial \underline{e}_4}{\partial \underline{x}_f} + \nu_5 \frac{\partial \underline{e}_5}{\partial \underline{x}_f} + \nu_7 \frac{\partial \underline{e}_7}{\partial \underline{x}_f} \tag{2.38}$$

Solving these equations requires knowledge of the values of the covector. Unfortunately, these values are unknowns.

F. COMPUTER PROGRAMMING

This section is intended to familiarize the reader with the mathematical model developed to solve of this problem. In order to utilize DIDO, the problem formulation must first be fed into it in the proper way. If the problem is not formulated and coded properly, the solution will not be correct as DIDO will be solving a different problem altogether. A great deal of the focus of this work has gone into the development and validation of the model.

1. Model Components for DIDO

In order to properly code the problem for DIDO, five MATLAB m-script files are created to define various components of the problem formulation. First, the cost function is defined in a cost m-file. This file contains the definition of the cost function which was developed in Eq. (2.15). Second, the dynamics of the problem is defined. This is done in an m-file which contains all of the dynamic equations developed in Section B. A third file needs to be created which states the end-point constraints based on the problem boundary conditions. This file is referred to as the events file. The events ensure that the spacecraft starts at Earth with the desired escape velocity and ends at an asteroid within the desired rendezvous tolerances. The asteroid selection logic is also included in this file. The fourth file is the path file and it is where the constraints along the path of the spacecraft are enforced. The path constraints are defined at the nodes based on the primal states and controls.[33] The fifth file is the main file that calls the other four. This main file defines any constants and scaling factors and calls DIDO. The main file is necessary for the generation of all the necessary inputs for DIDO, such as the specifications of the nodes, bounds on the variables and the initial guesses. DIDO then takes all of the provided data and runs a search for an optimal control solution. When the model is properly formulated, and the

[33] I. M. Ross and F. Fahroo, “User’s Manual for DIDO 2001.”

problem statement accurately delineated, DIDO will provide the optimal solution. All of the files mentioned above are provided in Appendix B.

2. Scaling

It has already been stated that this is a low-thrust optimization problem, as such, the thrust force is quite small. The distances in question, on the other hand, are quite large. The problem is being solved numerically through the use of an embedded iterative optimization program. This introduces very important factors for consideration, round-off and truncation errors. Due to the large differences in the magnitudes of the values being used in this problem, it is necessary to scale all of the variables so that they will have equal importance in the analysis. The use of canonical units can be implemented to normalize the values of the variables and greatly diminish the effects of round-off error on the final solution. A common "Distance-Unit" (DU), "Time-Unit" (TU), "Mass-Unit" (MU), "Velocity-Unit" (VU), and "Force-Unit" (FU) can be generated based on the scale of the problem at hand.[34] This form of units is extremely valuable in solving the problem using a numeric approach. The variables must, however, be related through given values. An example of the development of a set of canonical units is shown below.

$$\text{Let DU}=10\cdot\text{AU} \quad \text{and} \quad \text{Let MU}=1000\text{ kg}$$

$$TU=\sqrt{\frac{DU^3}{\mu}} \text{ based on Kepler's 3}^{\text{rd}}\text{ law} \tag{2.39}$$

$$FU=\frac{\mu\cdot MU}{DU^2} \text{ based on Newton's Law of Graviation} \tag{2.40}$$

These units can be developed further into other values such as the velocity-unit (VU=DU/TU). It is important to balance all equations so that all of the variables used are equally scaled. Great difficulty can come from poorly scaled or

[34] Jerry Jon Sellers, *Understanding Space*, 699-700.

balanced equations.[35] If proper care and attention is paid to the scaling of a problem these difficulties can be avoided and the problem will be significantly simplified.

G. FEASIBILITY OF SOLUTION

The solution found by DIDO should be checked for both optimality and feasibility. In order to verify the feasibility of the solution, the control trajectory produced by DIDO should be propagated through the dynamics of the problem. The propagated states at the nodes should match the values generated by DIDO. These values can be plotted together to visually check the feasibility of the solution provided by DIDO, i.e., if the propagated solution passes through the nodes the solution is visually feasible. DIDO calls for the specification of the number of nodes that the solution will be based on. As a first guess, to speed the solution process, 15 nodes were used to solve the problem. This result was then inserted into DIDO (bootstrapped) as a guess for a solution with 30 or more nodes. This higher number of nodes increases the accuracy of the result allowing for better analysis of the results. In order to verify that the plot of the states was generated based on the proper propagated control solution, the interpolated control variables that are used for state propagation are checked against their node point values from DIDO as well. Figure 1 and Figure 2 illustrate this consideration. It is important to note that a feasible solution in no way means an optimal solution has been generated. Rather, it only means that the state/control solution generated by DIDO satisfies the dynamics of the problem.

To conclude the feasibility analysis the end-point constraints of the problem must also be checked (in addition to the visual check) in order for a solution to be feasible. A sample of a feasible solution is provided in Figure 1 and a sample of an infeasible solution is provided in Figure 2.

[35] I. M. Ross and F. Fahroo, “User’s Manual for DIDO 2001.”

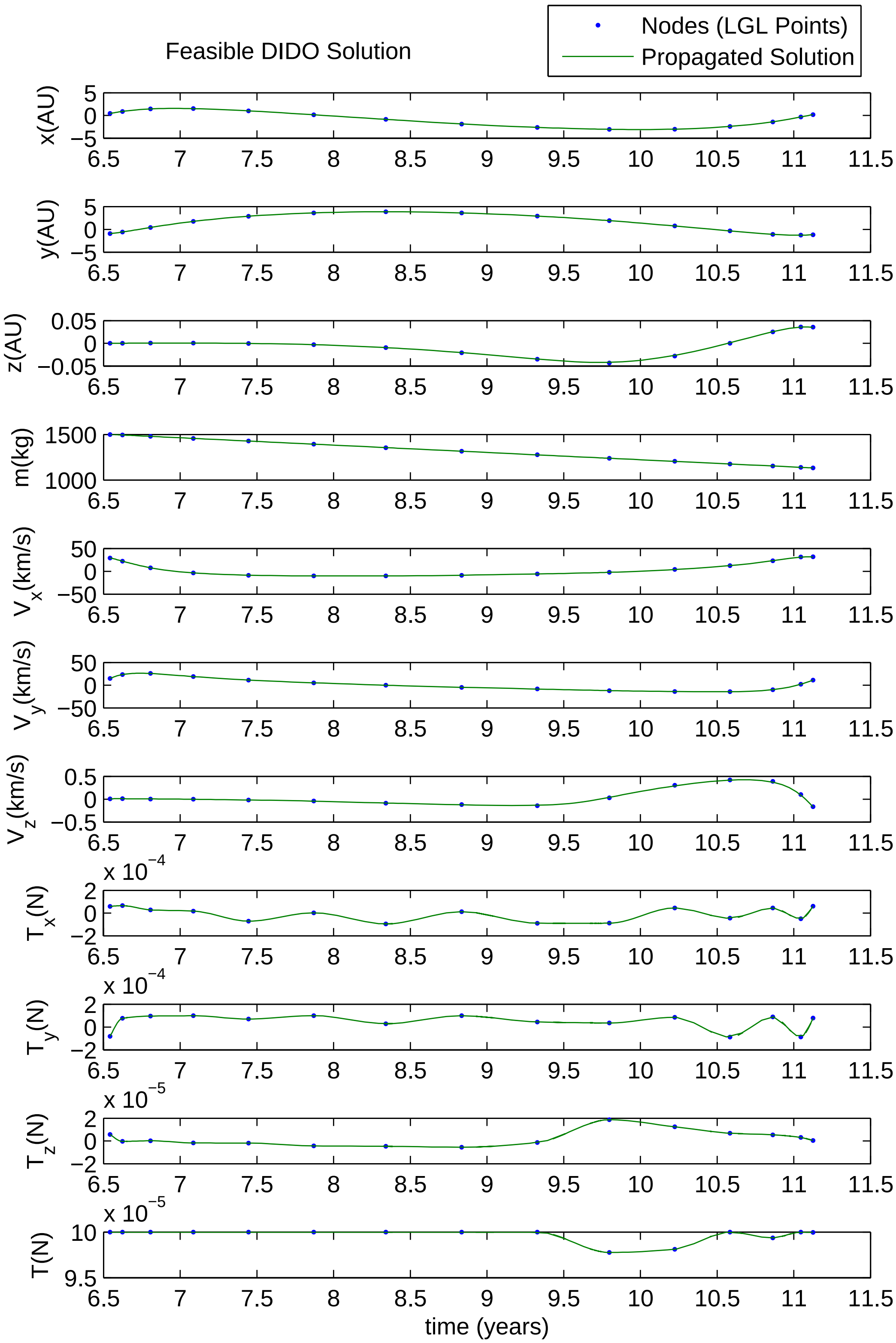

Figure 1: Sample Feasible DIDO Solution

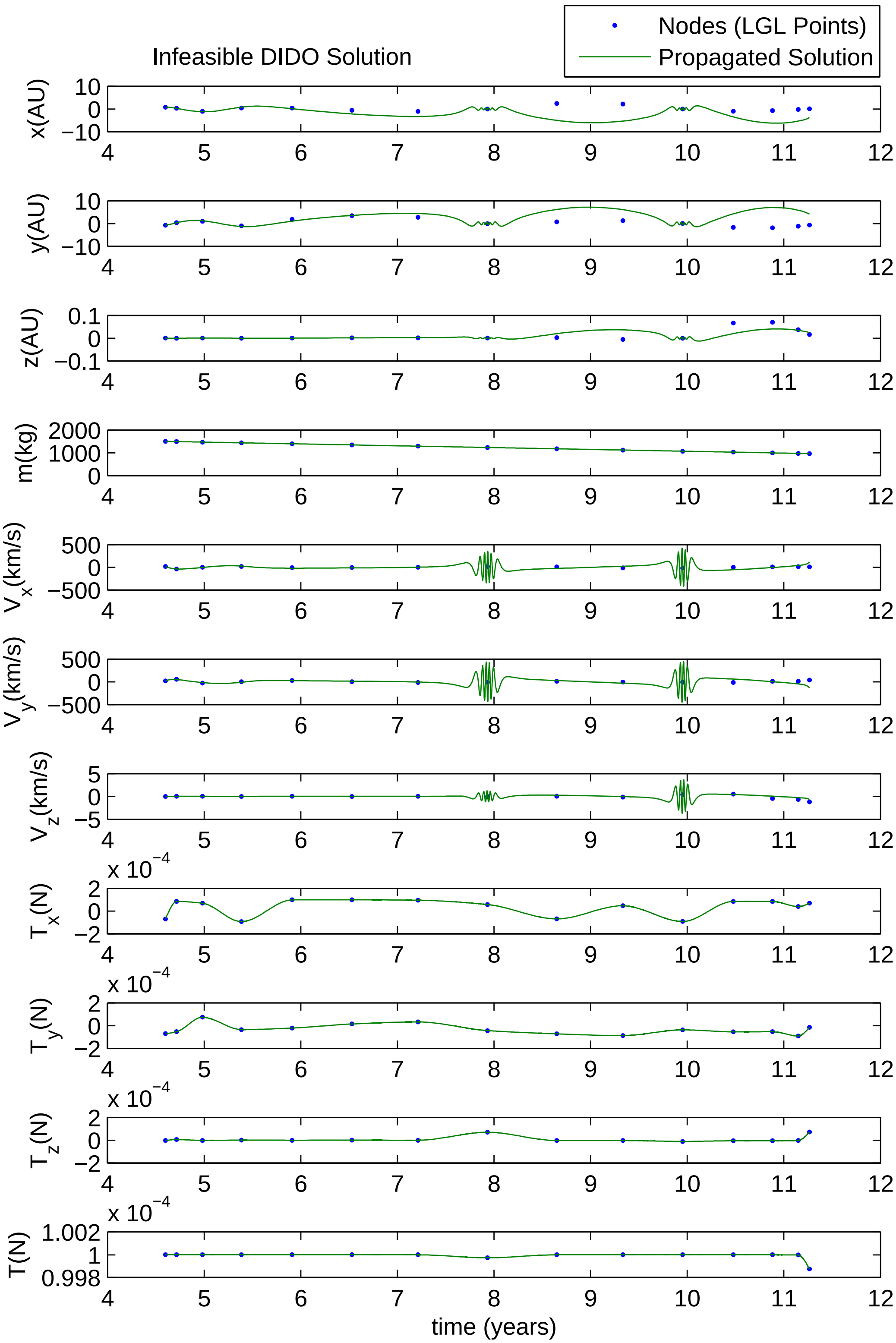

Figure 2: Sample Infeasible DIDO Solution

THIS PAGE INTENTIONALLY LEFT BLANK

III. ONE-TO-ONE TRANSFERS

A. INTRODUCTION

The purpose of this work is to select the optimal trajectory to rendezvous with an asteroid, given a group of asteroids. The selected solution must be verified as the optimal. In order to verify the selection, it is necessary to determine the cost for the spacecraft's one-to-one rendezvous with each asteroid. These costs can be compared, the minimum determined, and this minimum value becomes the basis for optimality for the selection verification. This section summarizes the results of the one-to-one spacecraft-asteroid rendezvous.

B. ONE-TO-ONE RENDEZVOUS RESULTS

The first step taken in this work was to determine the cost for a one-to-one rendezvous between the earth and a single asteroid. This analysis is conducted for a reasonable number of asteroids (five) for comparison. The selection criteria would then be introduced over this group of five asteroids such that the selected asteroid and subsequent trajectory could be compared to the baseline established with the one-to-one transfers.

To execute a one-to-one rendezvous, the model of the system simply includes the Sun, the Earth, the spacecraft and the particular asteroid in question. The dynamics of these bodies is inputted into DIDO to produce a solution based on the boundary conditions specified. An initial test was run with 15 nodes on the final five asteroids in the data set. These five asteroids were chosen without any specific intent or special interest. The results of this test are shown in Table 4. The cost is displayed in canonical units (mass units per time unit). From the tabulated values, it is clear that the minimum cost for one-to-one transfer occurs between Earth and asteroid number 910.

C. VERIFICATION OF THE RESULTS

1. Feasibility Analysis

The feasibility plots for these one-to-one rendezvous are shown in Figure 3 through Figure 7. The 15 node solution initially generated was bootstrapped as a guess to create a 60 node solution with greater accuracy. The results of the 60 node solutions are shown in Table 5.

Table 4: One-to-One Transfer Costs and Visible Feasibility

Asteroid Number	One-to-one Cost	Visibly Feasible
906	0.15492	Yes
907	0.20091	Yes
908	0.094657	Yes
909	0.14122	Yes
910	0.08575	Yes

The minimum cost is once again found to be in a trajectory rendezvousing with asteroid 910. These costs have a higher level of confidence as they are obtained using more nodes. The increase accuracy is evident in the feasibility plots shown in Figure 8 through Figure 12.

Table 5: Bootstrapped One-to-one Transfer Costs and Visible Feasibility

Asteroid Number	One-to-one Cost (MU/TU)	Visibly Feasible
906	0.1243	Yes
907	0.26218	Yes
908	0.16541	Yes
909	0.14527	Yes
910	0.067906	Yes

In order to verify that the plots of the states in Figure 3 through Figure 12 were generated based on the proper propagated solutions, the control variables were propagated and checked for their feasibility as well. This confirms that the states are being generated by the same control solution that DIDO generated. Although the 15 node solutions were found to be visibly feasible, there was room for improvement. This is evident from the plots of the feasibility of the control solution. In some of the plots, the propagated solution passes in the same general shape that the nodes create, but it does not directly pass through each one. This shows a lack in the number of nodes used to find the control solution. As such, a higher fidelity solution was necessary. The 60 node solution, bootstrapped from the 15 node solution, shows a much higher level of confidence. In Figure 8 through Figure 12, the propagated solution passes directly through every node, with high accuracy and little room for variation.

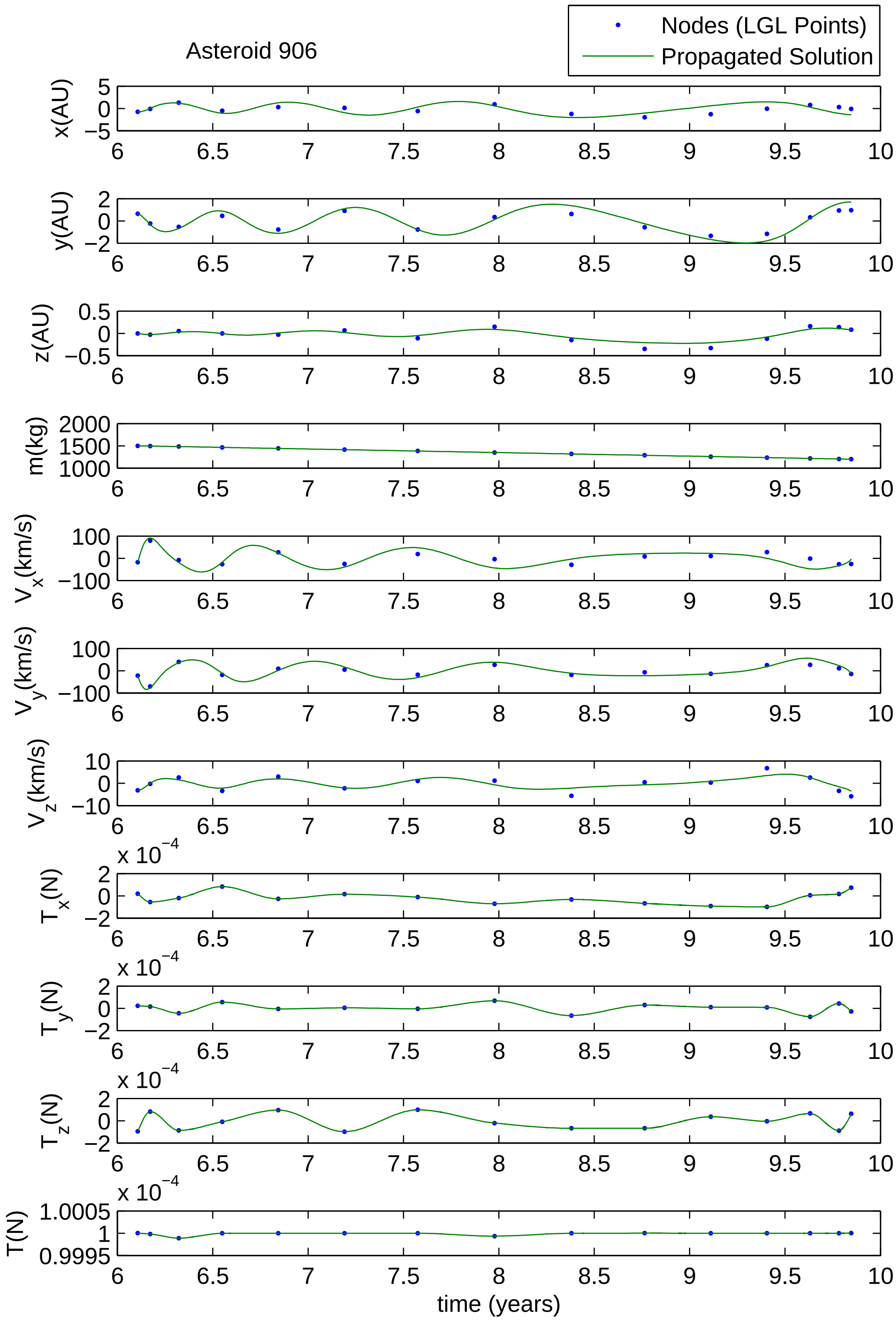

Figure 3: Visual Feasibility Plot for Asteroid 906 with 15 Nodes

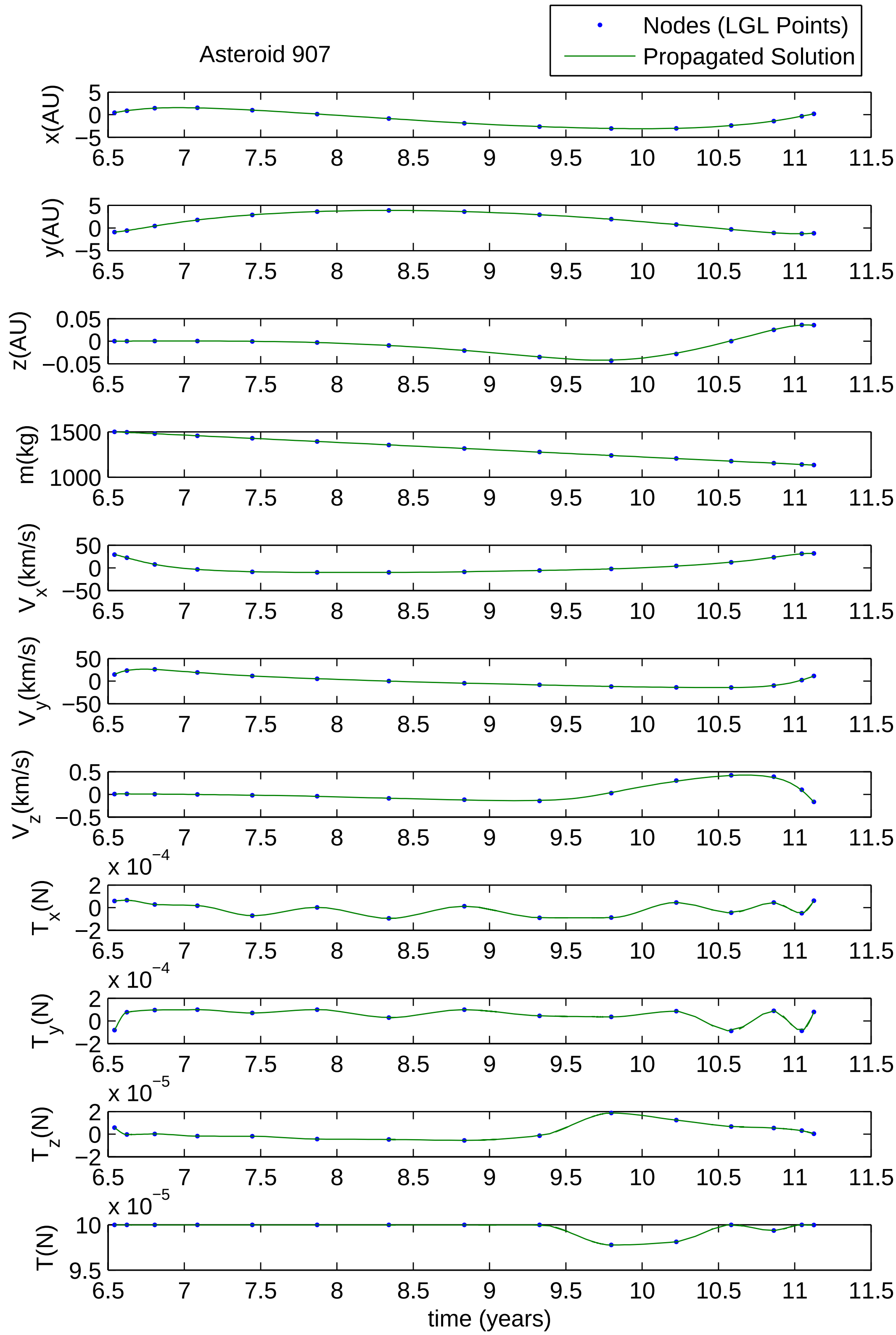

Figure 4: Visual Feasibility Plot for Asteroid 907 with 15 Nodes

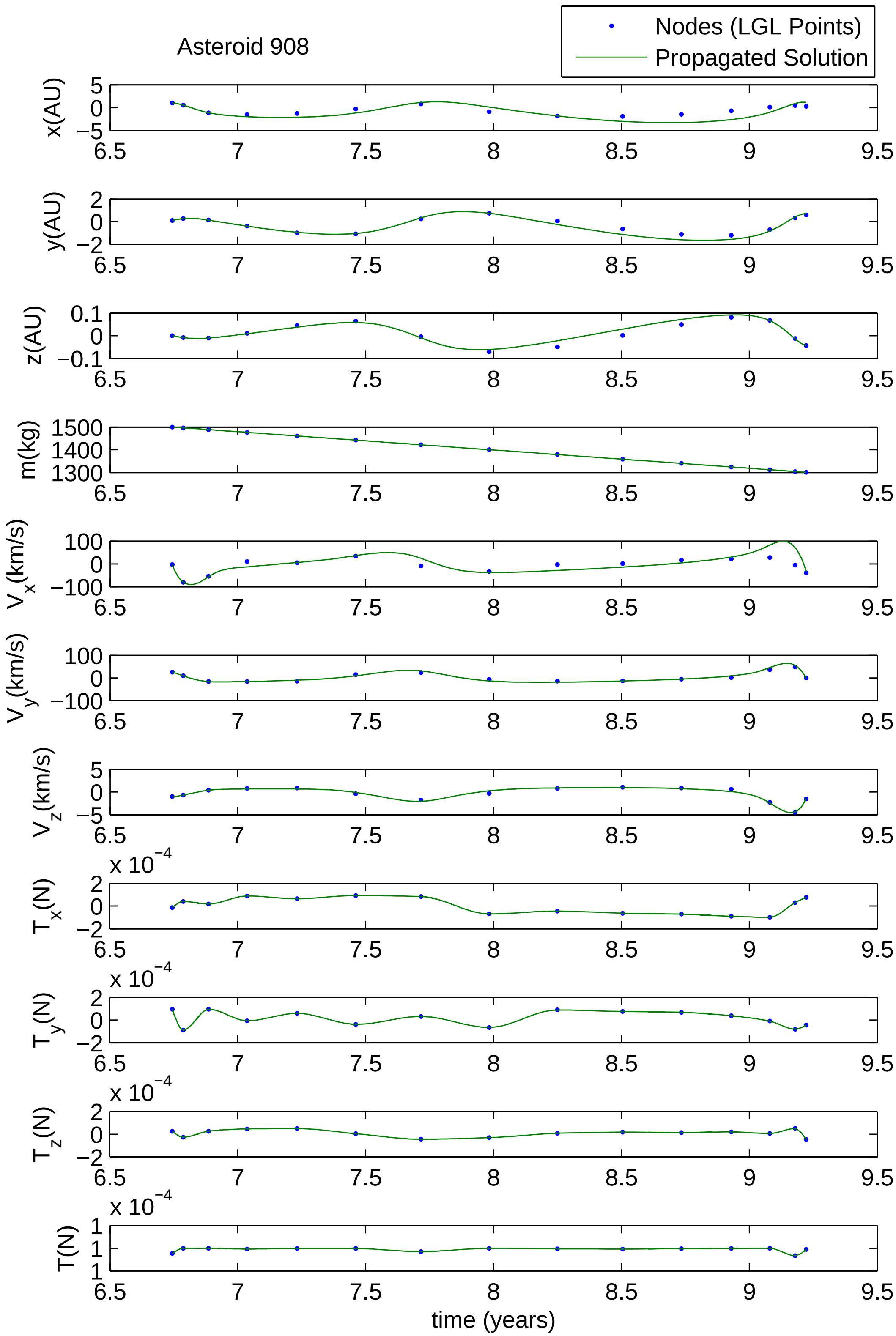

Figure 5: Visual Feasibility Plot for Asteroid 908 with 15 Nodes

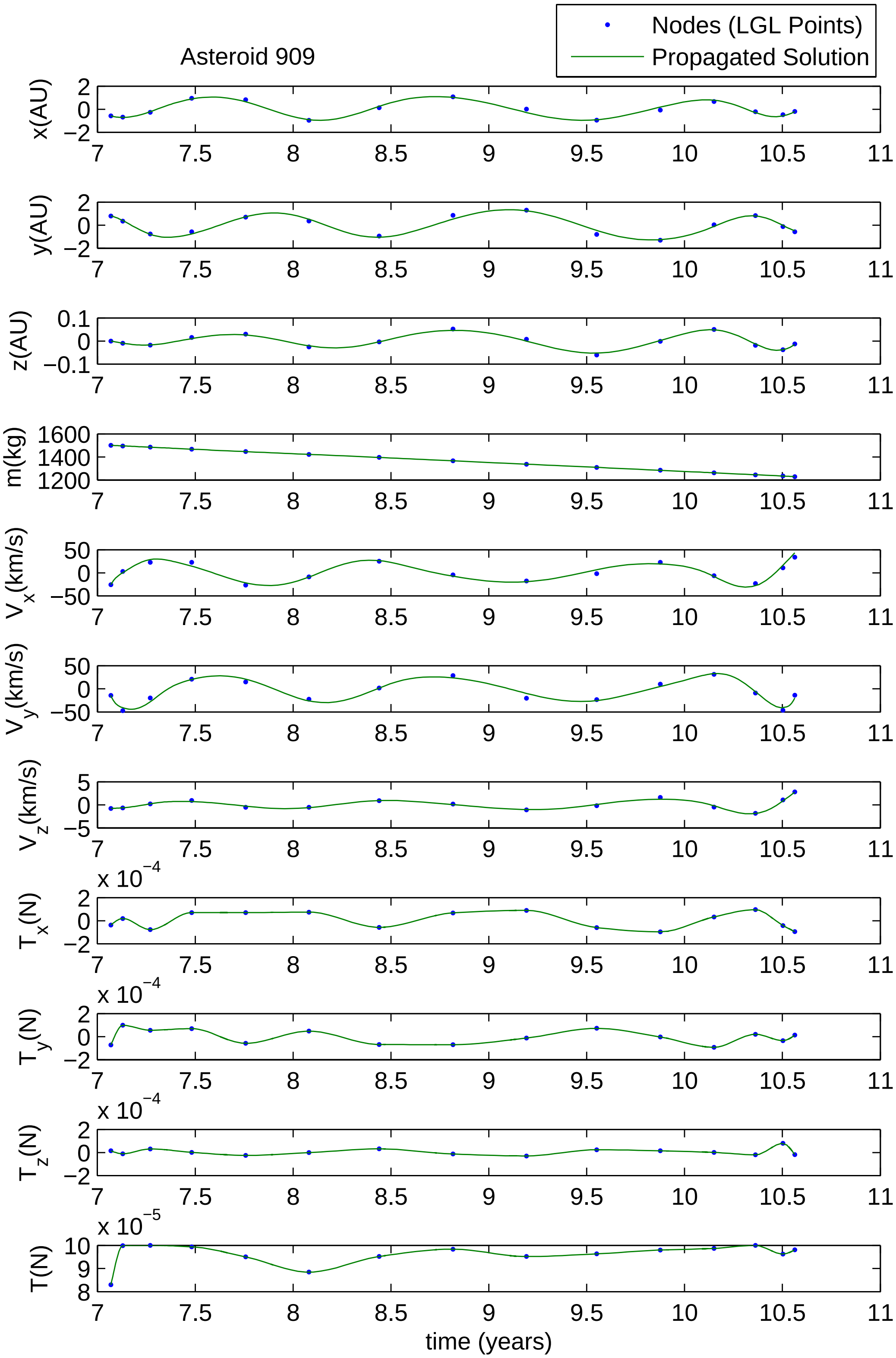

Figure 6: Visual Feasibility Plot for Asteroid 909 with 15 Nodes

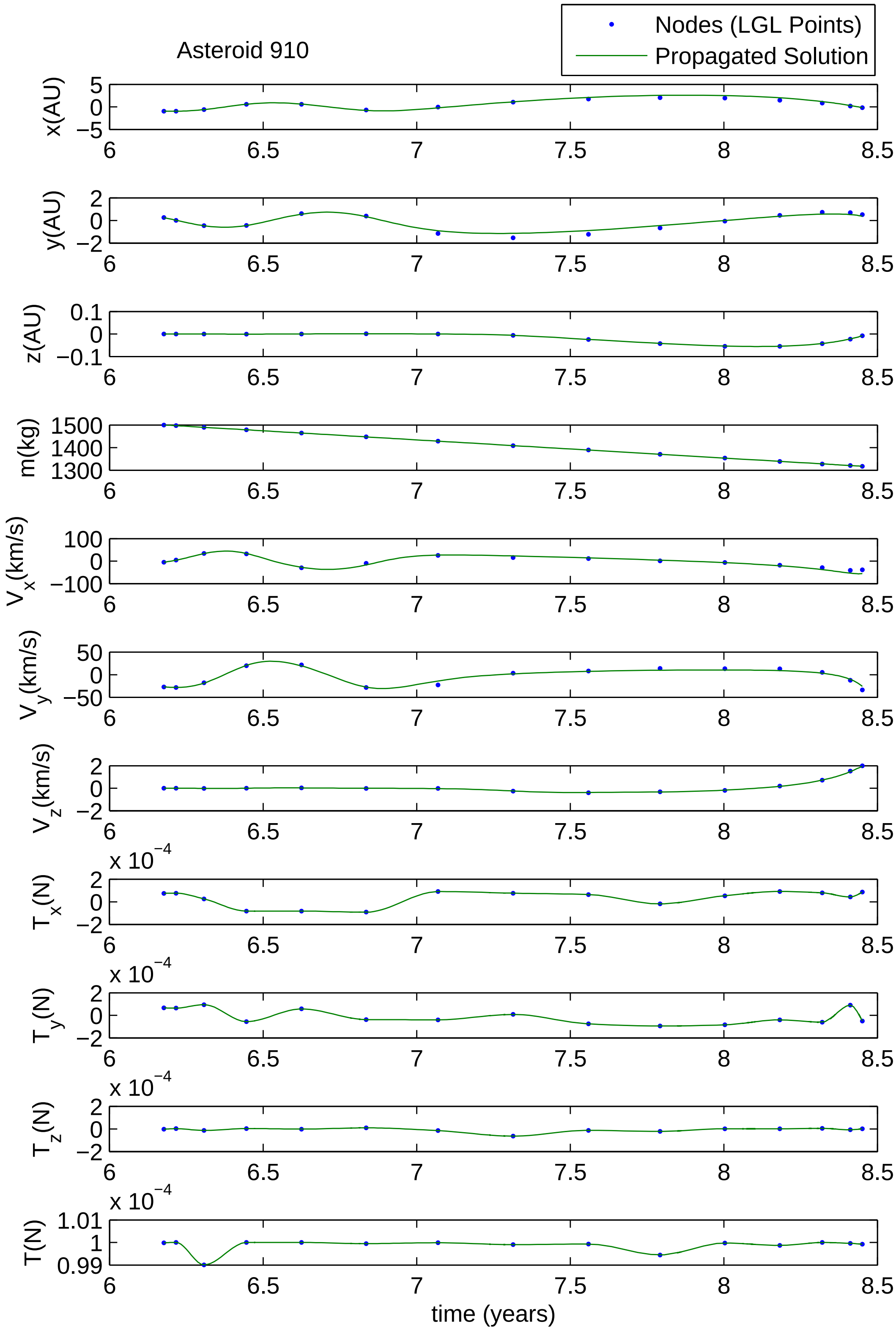

Figure 7: Visual Feasibility Plot for Asteroid 910 with 15 Nodes

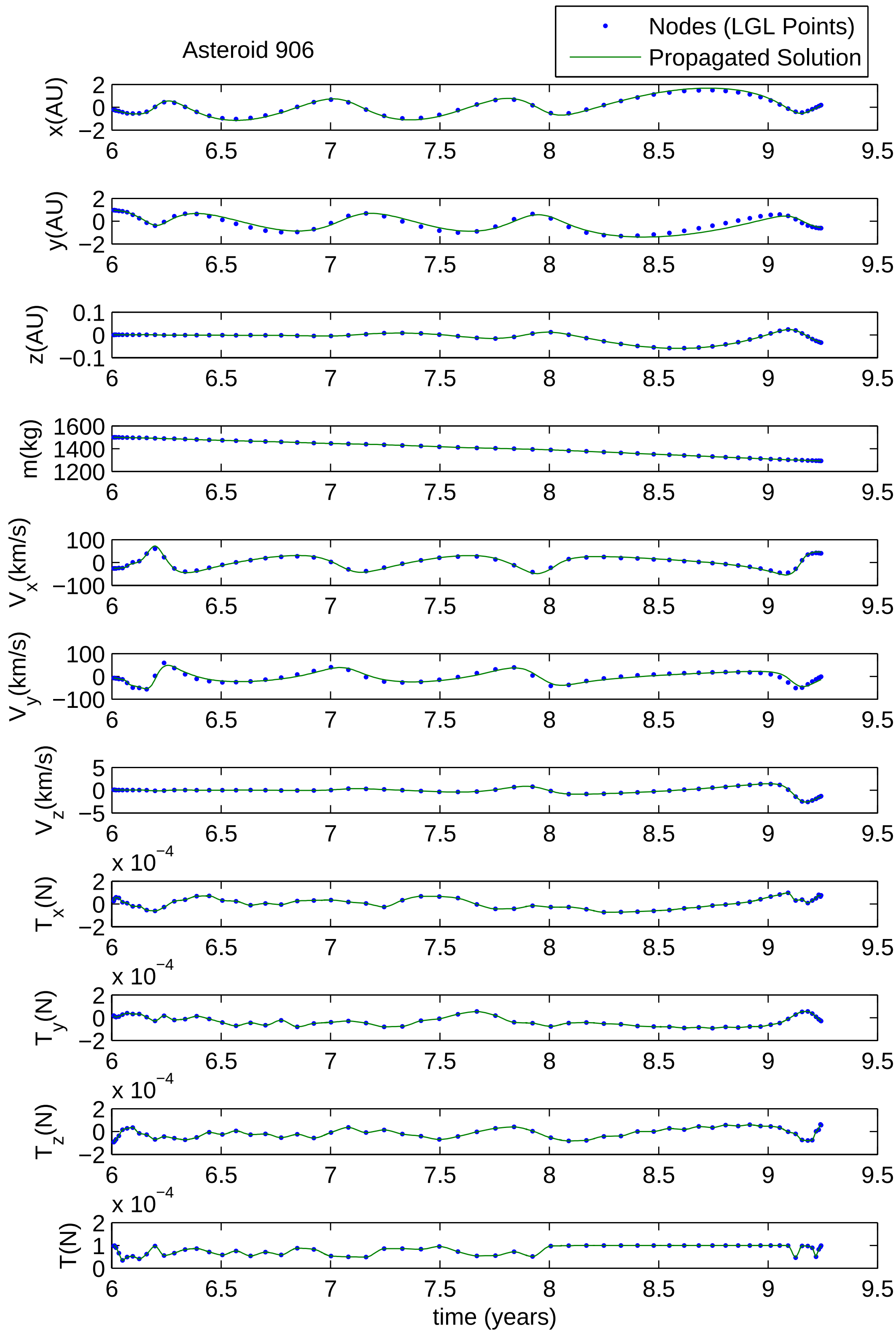

Figure 8: Visual Feasibility Plot for Asteroid 906 with 60 Nodes

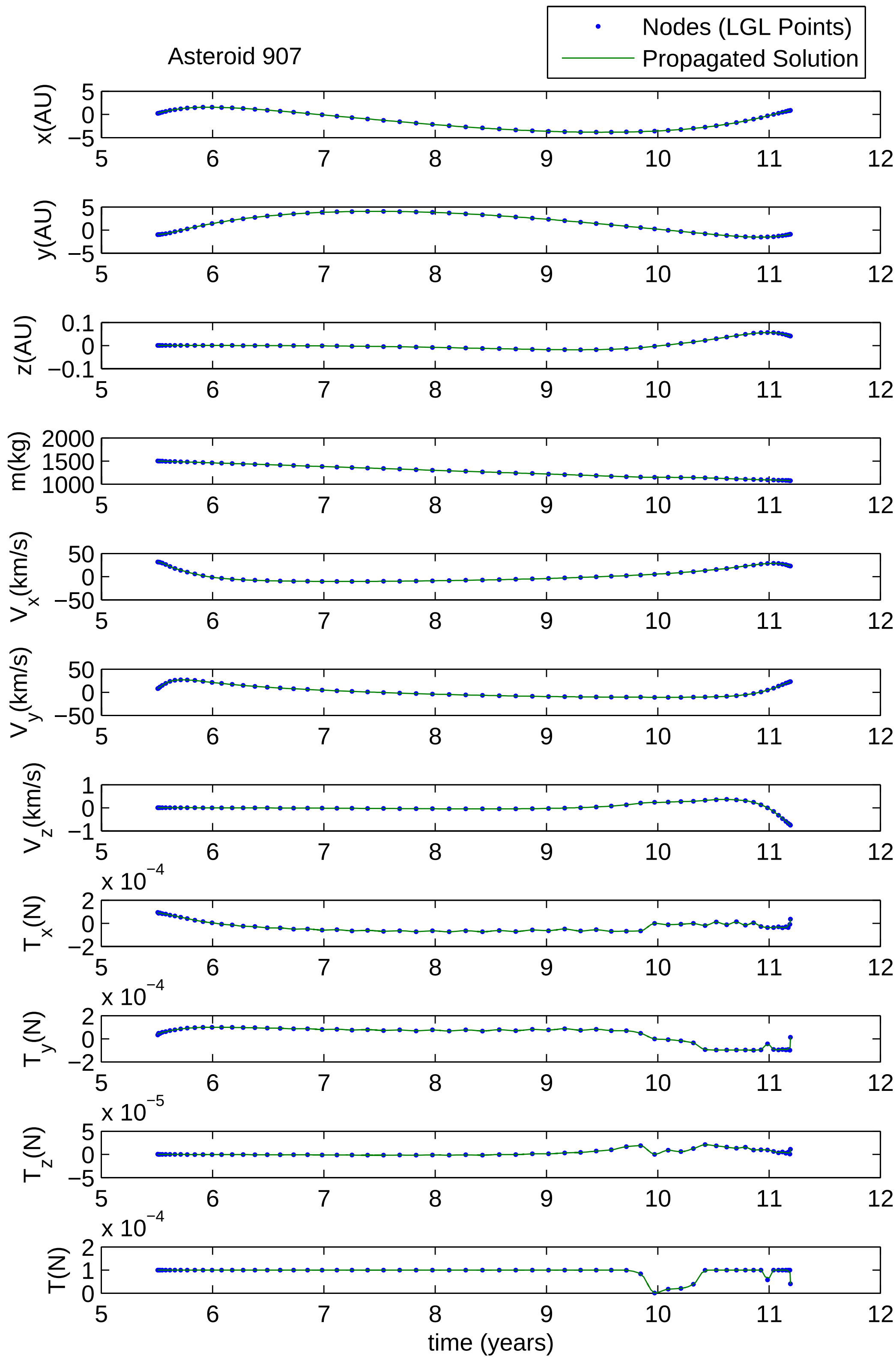

Figure 9: Visual Feasibility Plot for Asteroid 907 with 60 Nodes

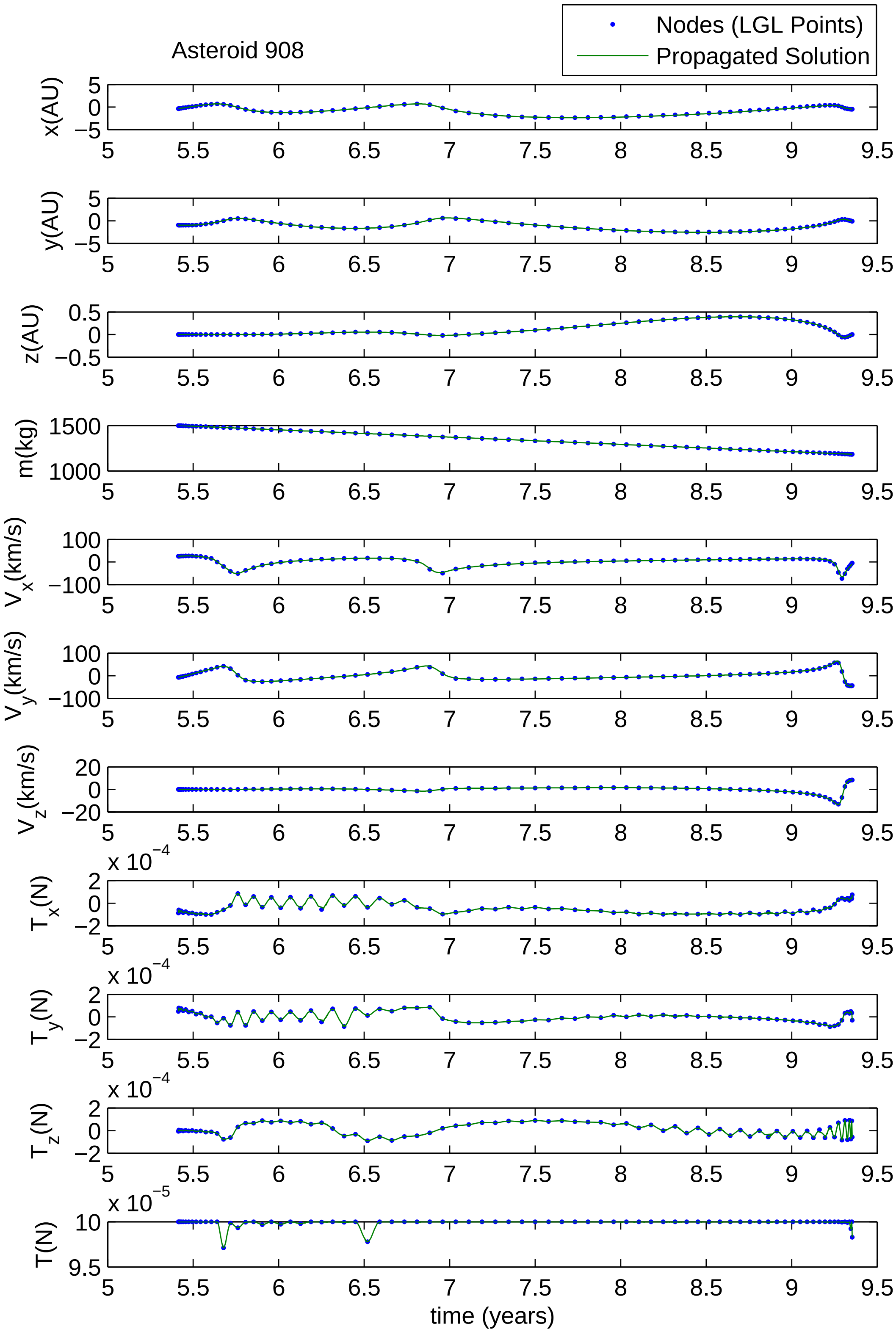

Figure 10: Visual Feasibility Plot for Asteroid 908 with 60 Nodes

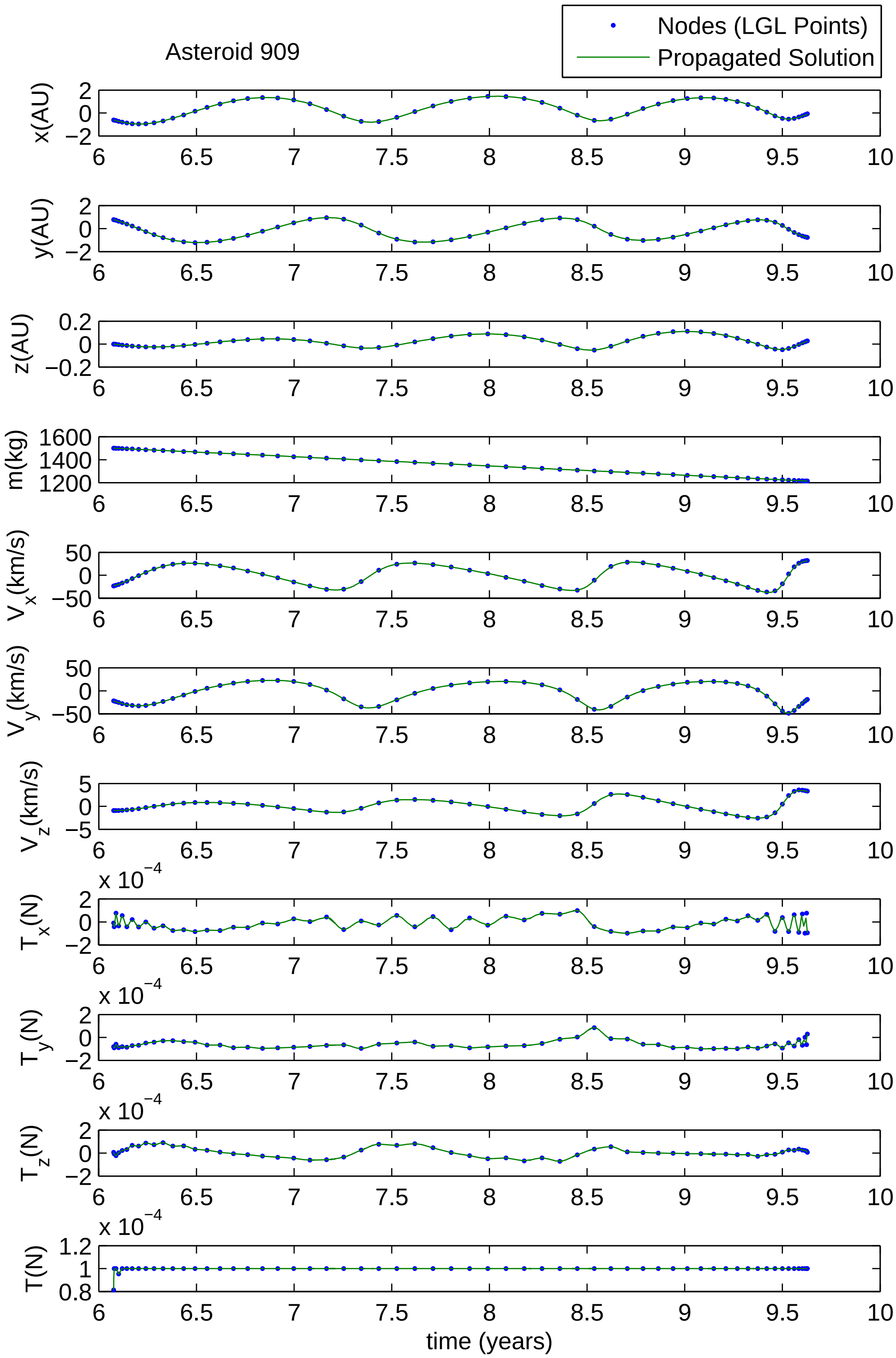

Figure 11: Visual Feasibility Plot for Asteroid 909 with 60 Nodes

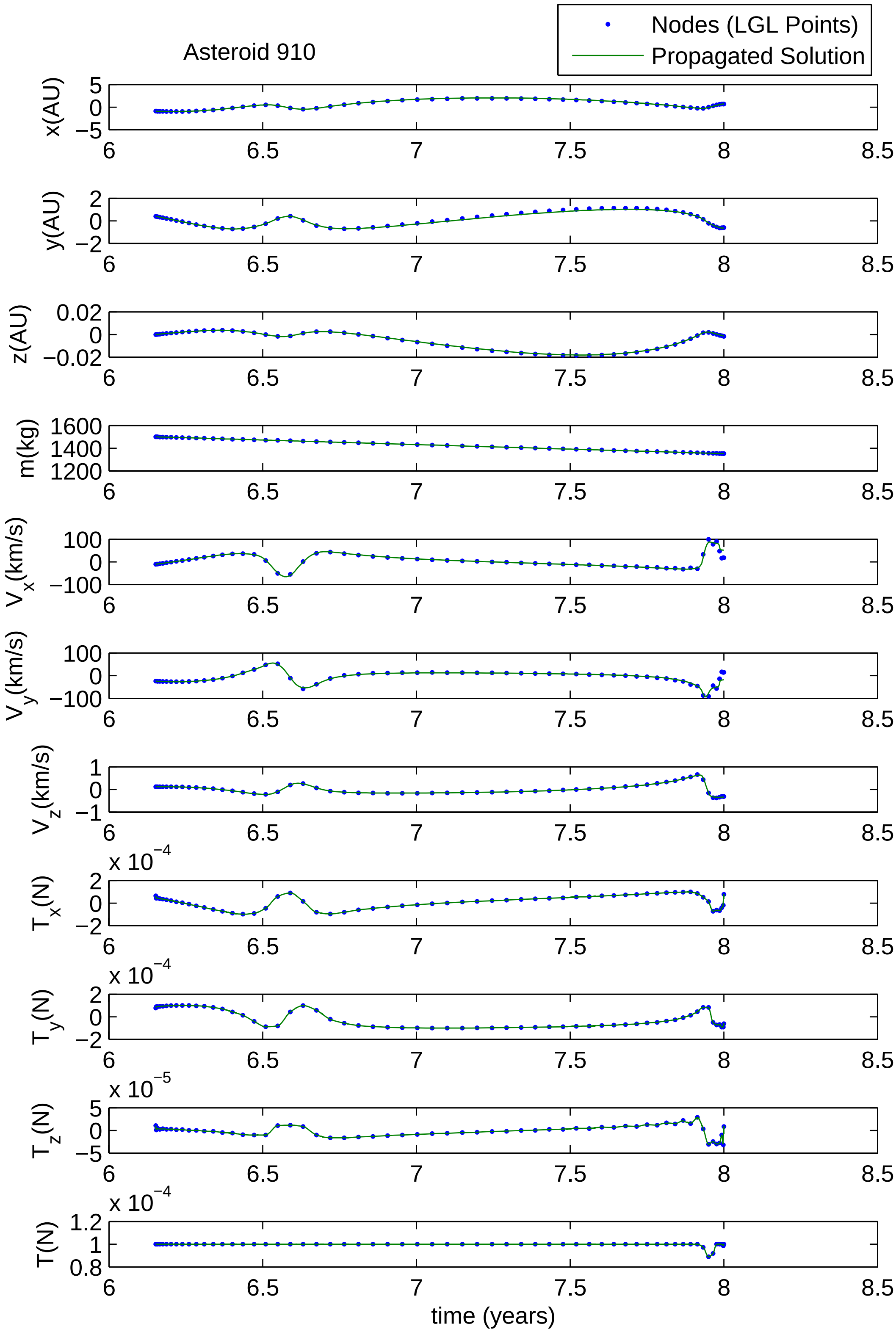

Figure 12: Visual Feasibility Plot for Asteroid 910 with 60 Nodes

Although Figure 8 through Figure 12 imply that the obtained solutions are visually feasible, more analysis must be conducted before the solutions can be confirmed as feasible. The solution must meet all of the boundary conditions set in the problem formulation (i.e. the spacecraft trajectory must start at Earth and end at the asteroid).

2. Checking the Boundary Conditions

DIDO outputs allow the user to easily check if the boundary constraints are met by the solution. The solution variables are all available and can be inserted into the events file to check if the end-points constraints are satisfied. The value of each constraint must fall between the limits established in the main file. There are seven of these end-point constraints: initial position, velocity, and mass of the spacecraft, final position and velocity of the spacecraft, mean anomaly of the Earth, and the asteroid. In order to confirm the feasibility of the solution, all of these constraints must be met.

The solutions generated in Section B do not meet the constraints on the final position and velocity of the spacecraft. These values are shown in Table 6. Table 6 displays the boundary conditions as they appear in the events file. The initial position takes the difference between the position of the spacecraft and the Earth at the first node.

$$\left(x_{S/C_0}^2 - x_{Earth_0}^2\right) + \left(y_{S/C_0}^2 - y_{Earth_0}^2\right) + \left(z_{S/C_0}^2 - z_{Earth_0}^2\right) \tag{3.1}$$

Initial velocity is the difference between the velocities of the spacecraft and the Earth at the first node. This value has a range because the spacecraft has an initial escape velocity.

$$\left(v_{x_{S/C_0}}^2 - v_{x_{Earth_0}}^2\right) + \left(v_{y_{S/C_0}}^2 - v_{y_{Earth_0}}^2\right) + \left(v_{z_{S/C_0}}^2 - v_{z_{Earth_0}}^2\right) \tag{3.2}$$

Spacecraft mass is the difference between the initial spacecraft mass and the initial mass generated by DIDO.

$$m_0 - m_{0_{calculated}} \tag{3.3}$$

Final position is the difference between the position of the spacecraft and the asteroid at the final node. This value has a range to allow for a rendezvous tolerance.

$$\left(x_{S/C_f}^2 - x_{Ast_f}^2\right) + \left(y_{S/C_f}^2 - y_{Ast_f}^2\right) + \left(z_{S/C_f}^2 - z_{Ast_f}^2\right) \tag{3.4}$$

Final velocity is the difference between the velocity of the spacecraft and the asteroid at the final node. This value also has a range to allow for a rendezvous tolerance.

$$\left(v_{x_{S/C_f}}^2 - v_{x_{Ast_f}}^2\right) + \left(v_{y_{S/C_f}}^2 - v_{y_{Ast_f}}^2\right) + \left(v_{z_{S/C_f}}^2 - v_{z_{Ast_f}}^2\right) \tag{3.5}$$

M_{Earth} refers to the mean anomaly of the Earth. It is calculated as the difference between the mean anomaly based on the given data and the calculated mean anomaly from DIDO all at the first node.

$$M_{Earth} - E_0 + e_{Earth} * \sin(E_0) \tag{3.6}$$

Similarly, M_{ast} refers to the mean anomaly of the asteroid and is calculated in the same way, only at the final node.

$$M_{Ast} - E_f + e_{Ast} * \sin(E_f) \tag{3.7}$$

Table 6: Boundary Conditions for the 60 Node Solutions

Asteroid	Initial Postion	Initial Velocity	Spacecraft Mass	Final Position	Final Velocity	M_{Earth}	M_{Ast}
Lower Limit	0	0	0	0	0	0	0
Upper Limit	0	122.5	0	4.47E-13	1.00E-10	0	0
906	8.84E-7	122.5002	0	0.0012	7.47E-6	-3.99E-14	-2.84E-9
907	5.54E-6	122.5000	1.87E-13	0.0013	3.75E-4	5.51E-13	-4.33E-7
908	5.94E-7	122.5000	0	2.06E-4	3.24E-6	-1.15E-12	1.49E-10
909	3.63E-6	122.5000	0	1.48E-4	8.13E-6	-1.55E-11	1.58E-11
910	5.29E-7	122.5000	4.89E-15	8.75E-4	7.32E-4	3.30E-14	6.66E-14

It is evident from this table that not all of the end-point constraints are met by these solutions. Many of the values are so small that the truncation and round-off errors inherent in numerical techniques are the cause of the variance. However, when the final position and velocity are observed more closely, there is an obvious disparity between the limit values and the values generated by the solution.

This disparity can also be demonstrated graphically by plotting the trajectory of the spacecraft and the orbits of the Earth and the asteroid. These plots are shown in Figure 13 through Figure 17 for the one-to-one rendezvous with asteroids 906 through 910 respectively. These figures all confirm that the spacecraft trajectory is not ending at the asteroid orbit. Thus, the solutions produced in Section B are confirmed to be infeasible. There is no need for an optimality analysis of infeasible solutions.

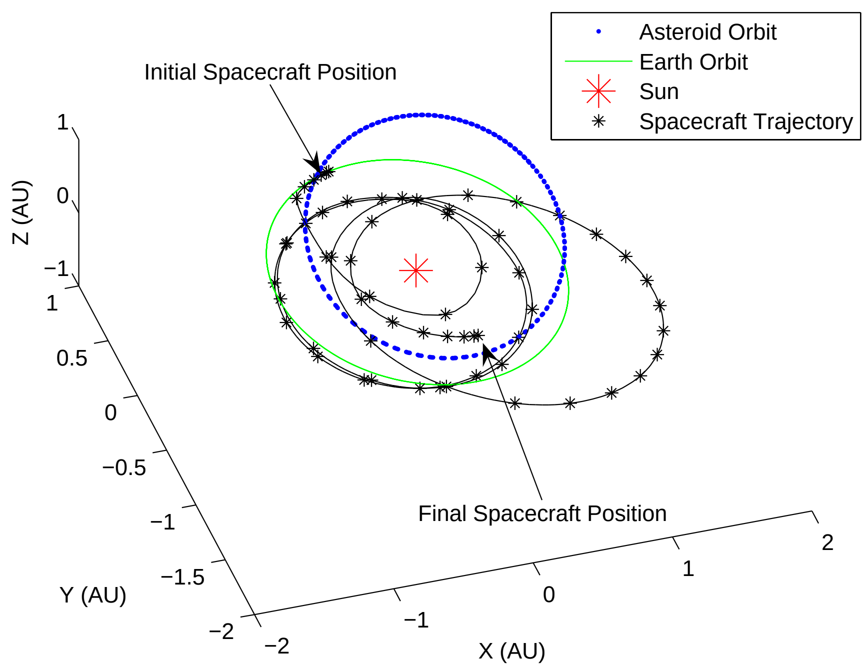

Figure 13: 3-D Plot of Spacecraft Trajectory with respect to the Earth and Asteroid 906 Orbits

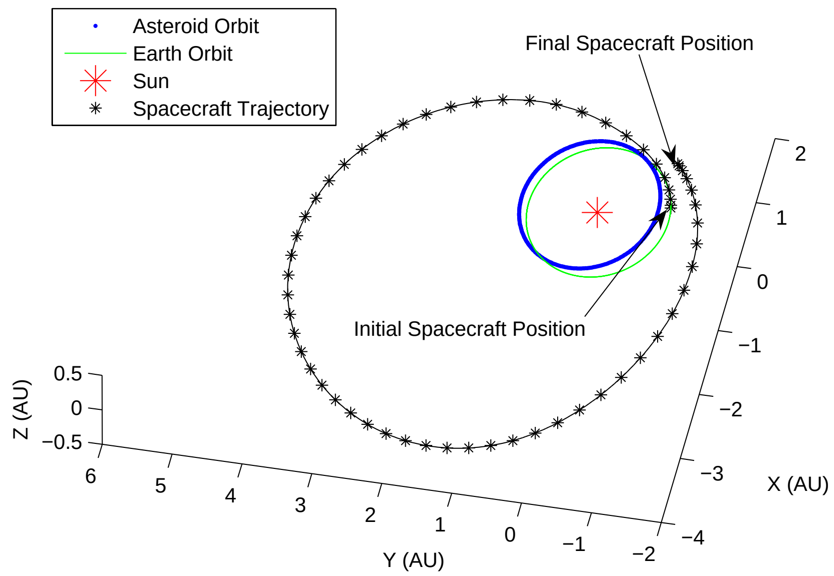

Figure 14: 3-D Plot of Spacecraft Trajectory with respect to the Earth and Asteroid 907 Orbits

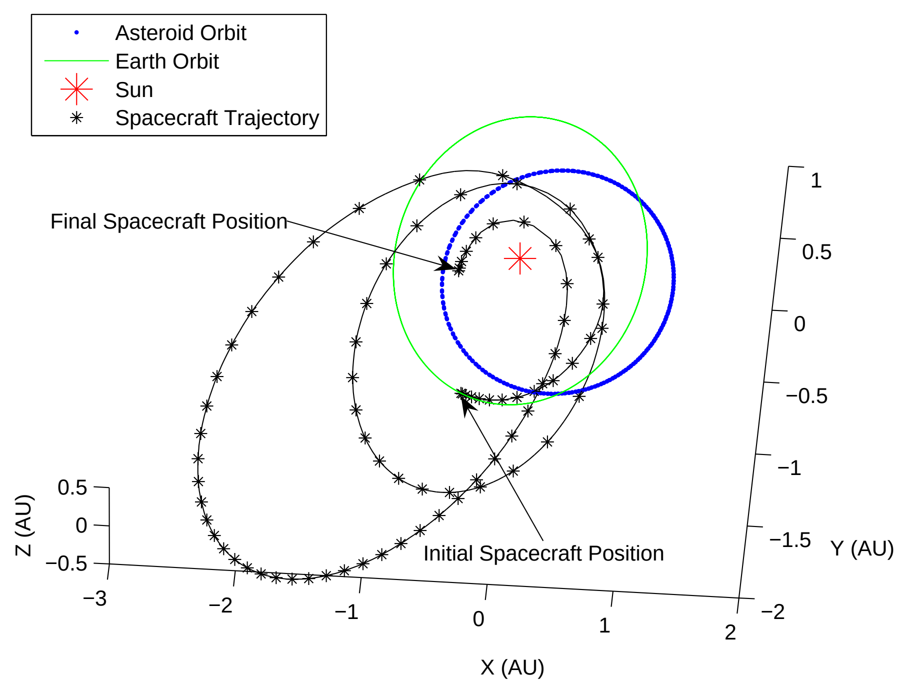

Figure 15: 3-D Plot of Spacecraft Trajectory with respect to the Earth and Asteroid 908 Orbits

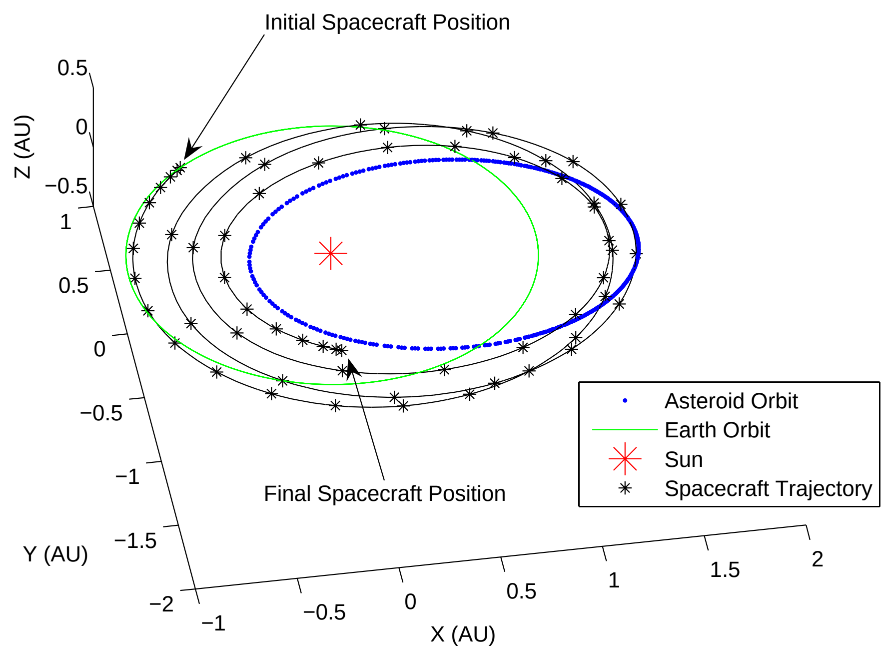

Figure 16: 3-D Plot of Spacecraft Trajectory with respect to the Earth and Asteroid 909 Orbits

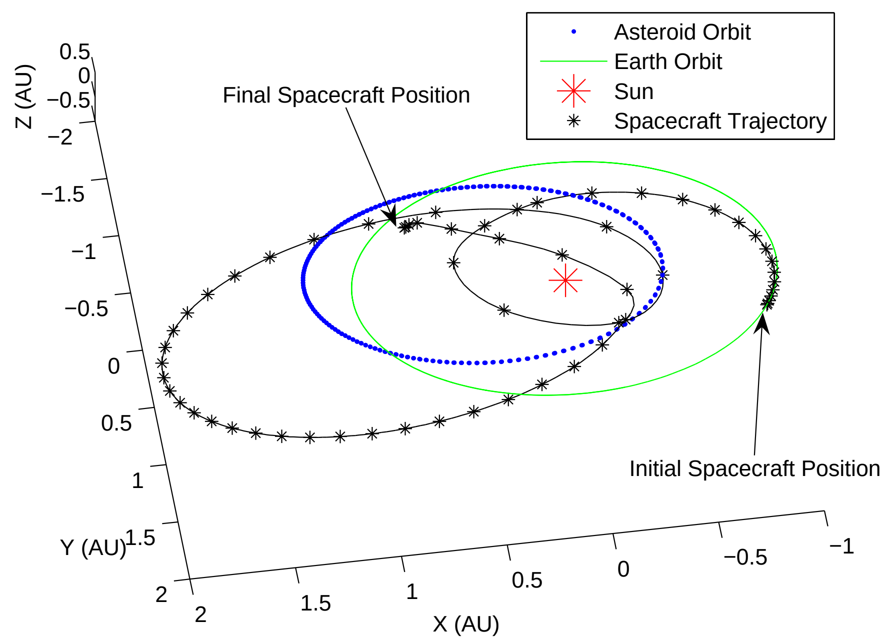

Figure 17: 3-D Plot of Spacecraft Trajectory with respect to the Earth and Asteroid 910 Orbits

D. CORRECTING THE MODEL

As a result of the analysis presented in Section C of this chapter, the solutions for the one-to-one spacecraft-asteroid rendezvous do not pass the feasibility and optimality tests. There must, therefore, be a problem with the model as it was coded. This section focuses on identifying the error and the steps taken to correct this error.

1. Problem Identification

When the model is suspected of containing errors, it is necessary to check every line of the code for typing and/or mathematical errors. If this line by line check fails to unearth the inaccuracies of the model, the dynamics and events files should be checked for accuracy. When the events file that was used to generate the solution in Section B was reviewed it became evident that the two variables which were falling outside of the bounding values (i.e. final position and velocity) were not scaled correctly. This is also evident in Table 6 as the numbers enforcing these constraints were on the order of 10^{-13} and 10^{-10} respectively. These values are so small they are not within the DIDO computation tolerance. In addition, any round-off or truncation error could cause a severe alteration in the outcome. These endpoint conditions must therefore be scaled such that the numbers enforcing them can produce a working solution.

2. Problem Correction

Having identified a possible source of the error in the model, the necessary steps to correct the problem were studied. The problem was re-scaled so that all of the constraints are working within reasonably scaled numbers, so as to avoid errors caused by round-off or truncation errors. Fortunately, only the two constraints pertaining to the two final values in question need to be scaled, as the other values are already within reasonable orders of

magnitude. These two constraint equations were, therefore, entirely multiplied by a common multiplicative factor which would raise their order of magnitude. Both the constraints desired bounds and the functions which generate the constraint values must be multiplied by the same factor. When properly executed, the orders of magnitude of these two constraints are raised, without altering the configuration of the problem and without a need to completely identify new canonical units.

This technique for attempting to correct the model's deficiencies was utilized for the same five asteroids. The scale factors were adjusted to obtain a solution that meets all of the end-point criteria. Based on the orders of magnitude before this constraint scaling, a good starting point could be established for bringing the numbers to a reasonable magnitude. Eventually, using a scaling factor of 1×10^{-8} for the final position value and a scaling factor of 1×10^{4} for the final velocity value, we obtained one-to-one rendezvous trajectories to asteroids 906 through 910 that meet all of the end-point criteria and have a feasible solution. The feasibility plots are shown in Figure 18 through Figure 22 and the resulting costs are given in Table 7. These solutions provide trajectories which hold the end-point values within the constraints. These values are shown in Table 8.

Table 7: Rescaled One-to-One Rendezvous

Asteroid Number	One-to-one Cost	Visually Feasible
906	0.1497	Yes
907	0.2115	Yes
908	0.1292	Yes
909	0.1369	Yes
910	0.0915	Yes

Table 8: Boundary Conditions for the Rescaled One-to-One Rendezvous

Asteroid	Initial Postion	Initial Velocity	Spacecraft Mass	Final Position	Final Velocity	M_{Earth}	M_{Ast}
Lower Limit	0	0	0	0	0	0	0
Upper Limit	0	122.5	0	0.01	0.01	0	0
906	6.27E-05	122.4954	0	0.0047	0.01005	5.09E-11	-3.09E-11
907	1.65E-07	122.5	0	0.0068	0.01003	4.50E-13	4.33E-13
908	4.64E-07	122.5	4.80E-08	0.01002	0.0085	-7.54E-14	4.99E-12
909	2.53E-06	122.4994	0	0.0099	0.0098	-1.25E-11	-1.06E-13
910	8.46E-06	122.5	-2.20E-13	0.0089	0.01	1.50E-11	2.26E-11

The plots of the trajectories show the spacecraft rendezvousing with the asteroid as desired. This is shown in Figure 23 through Figure 27. The data shown is only for the 30 node solution bootstrapped from an initial solution of 15 nodes. These solutions meet the conditions for feasibility.

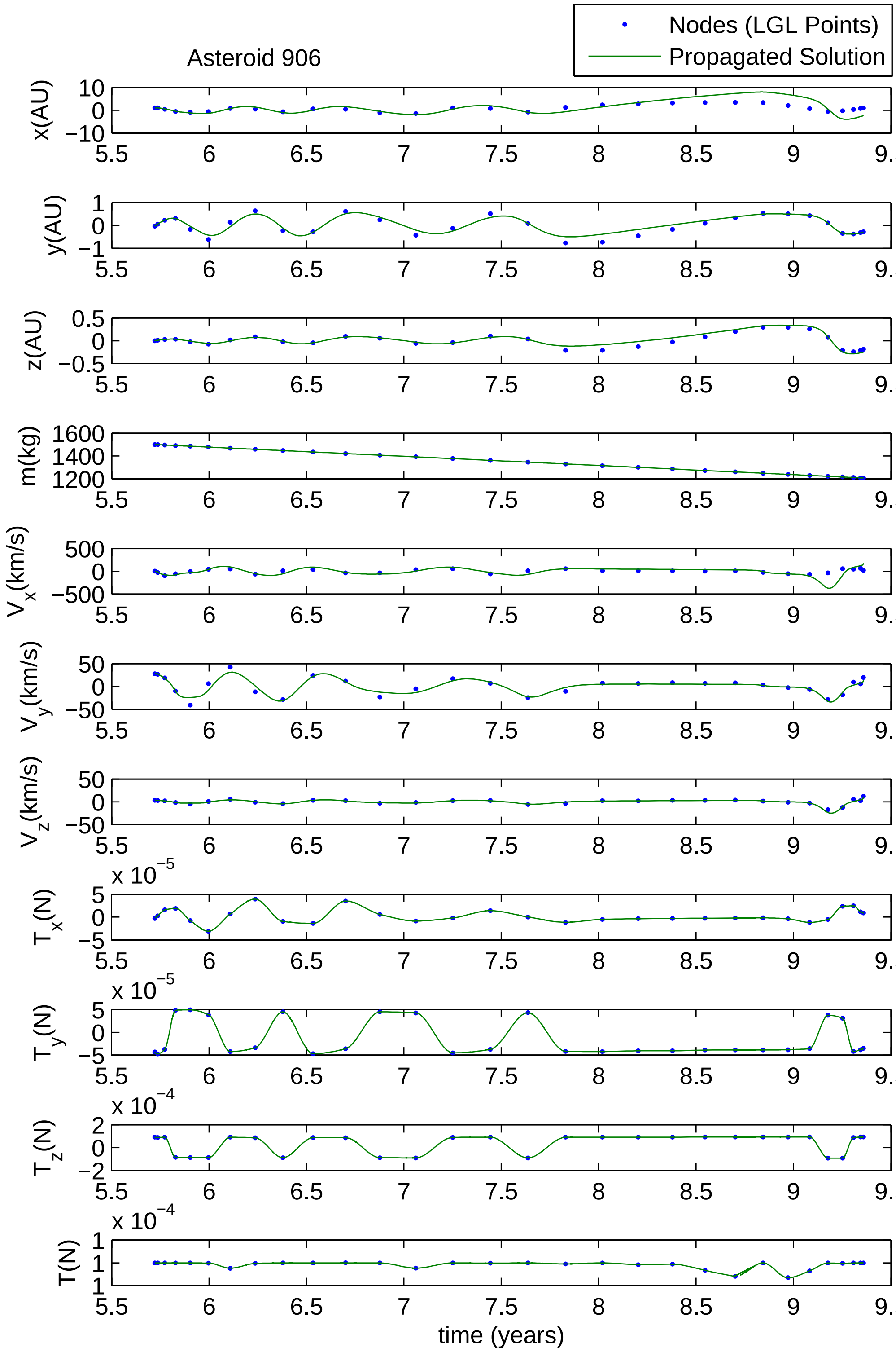

Figure 18: Rescaled 30 Node Visual Feasibility Plot for Asteroid 906

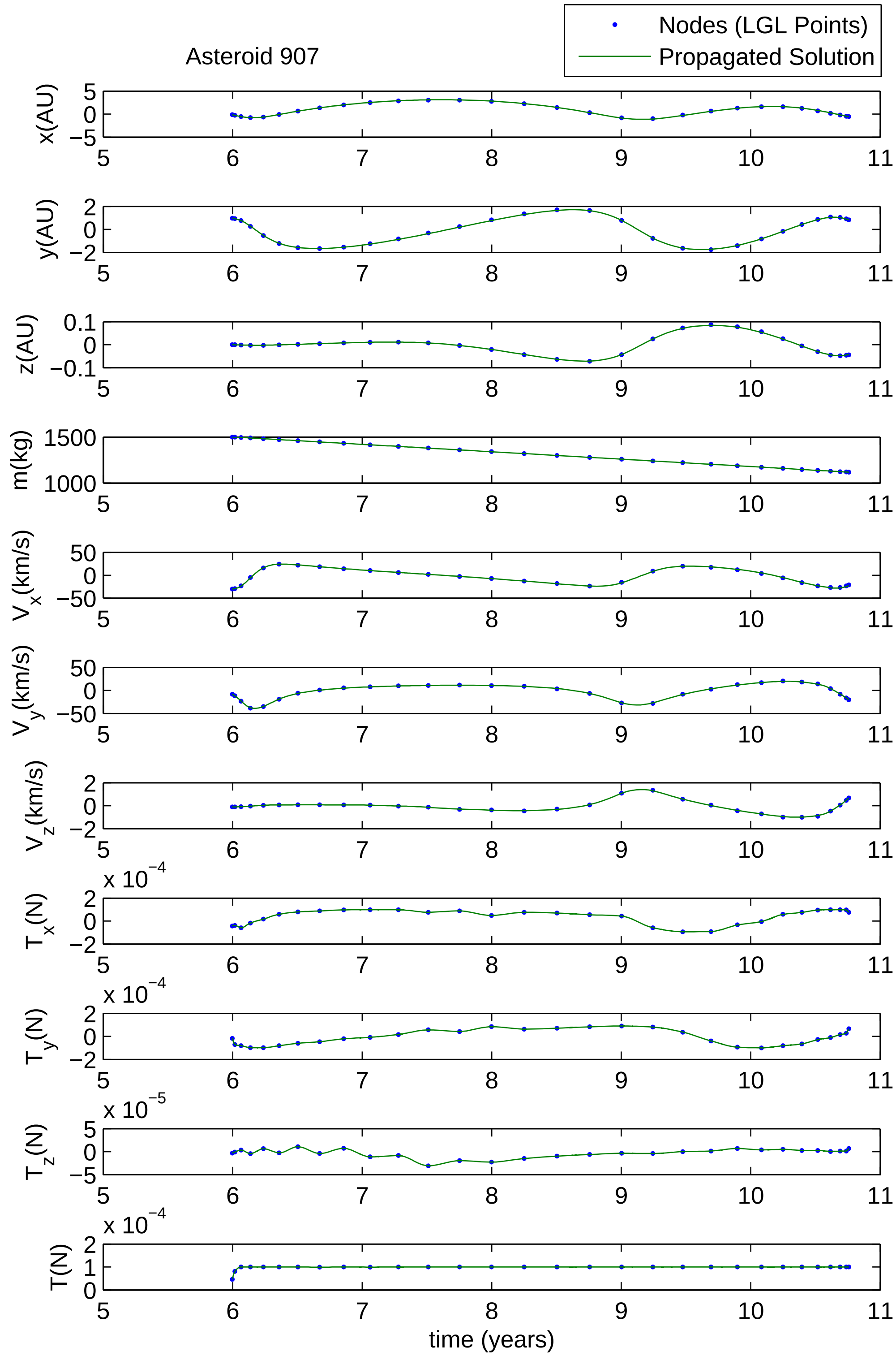

Figure 19: Rescaled 30 Node Visual Feasibility Plot for Asteroid 907

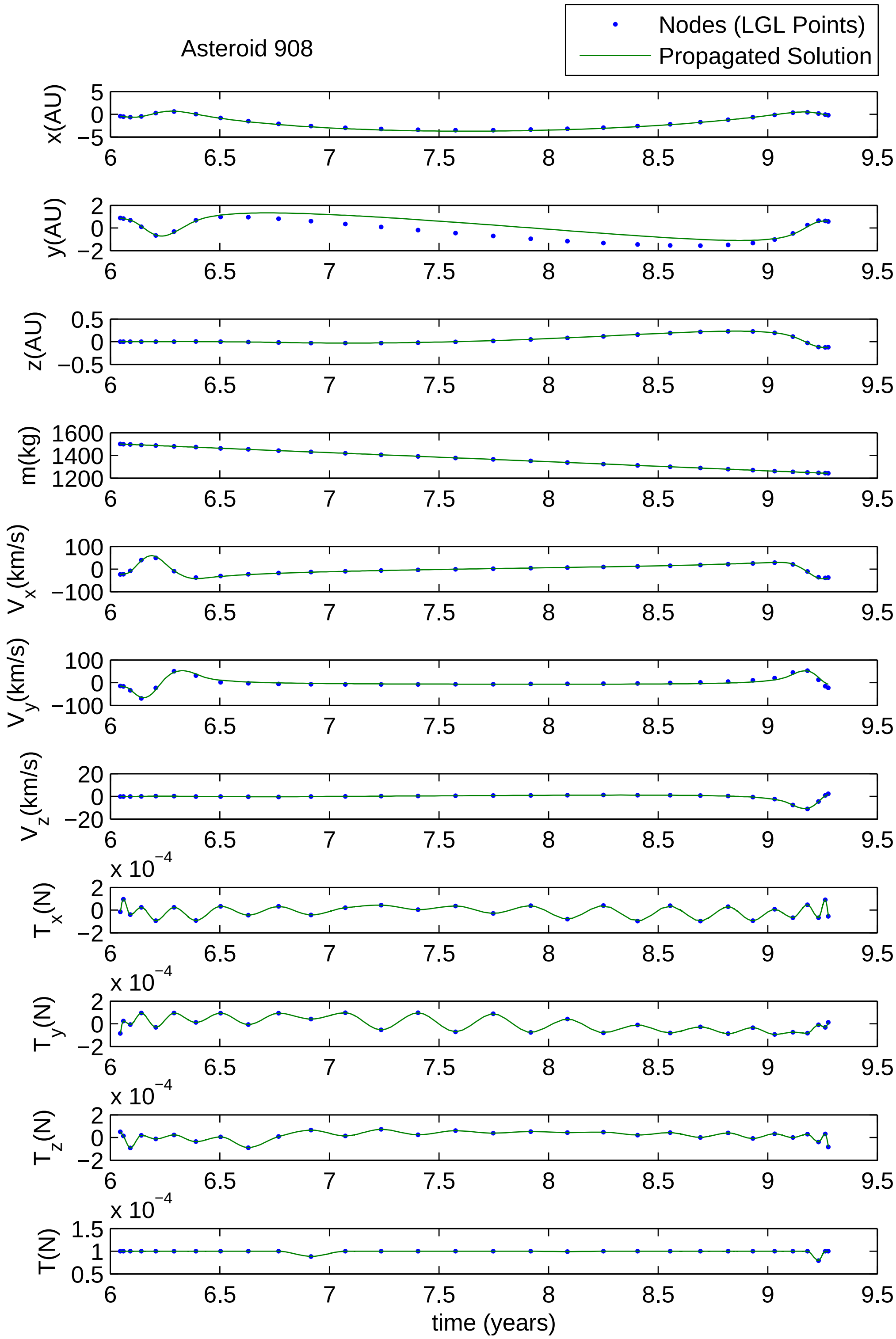

Figure 20: Rescaled 30 Node Visual Feasibility Plot for Asteroid 908

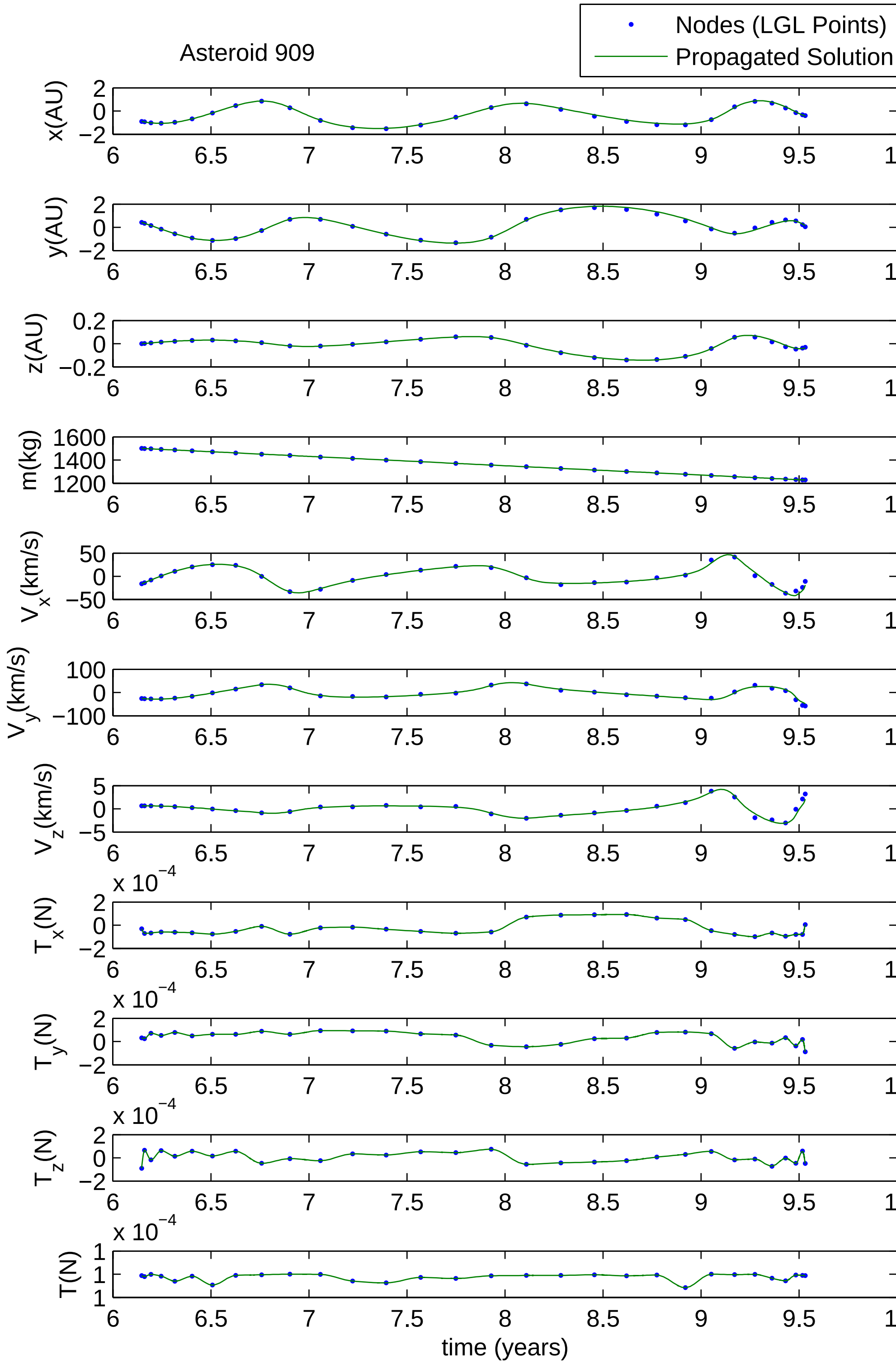

Figure 21: Rescaled 30 Node Visual Feasibility Plot for Asteroid 909

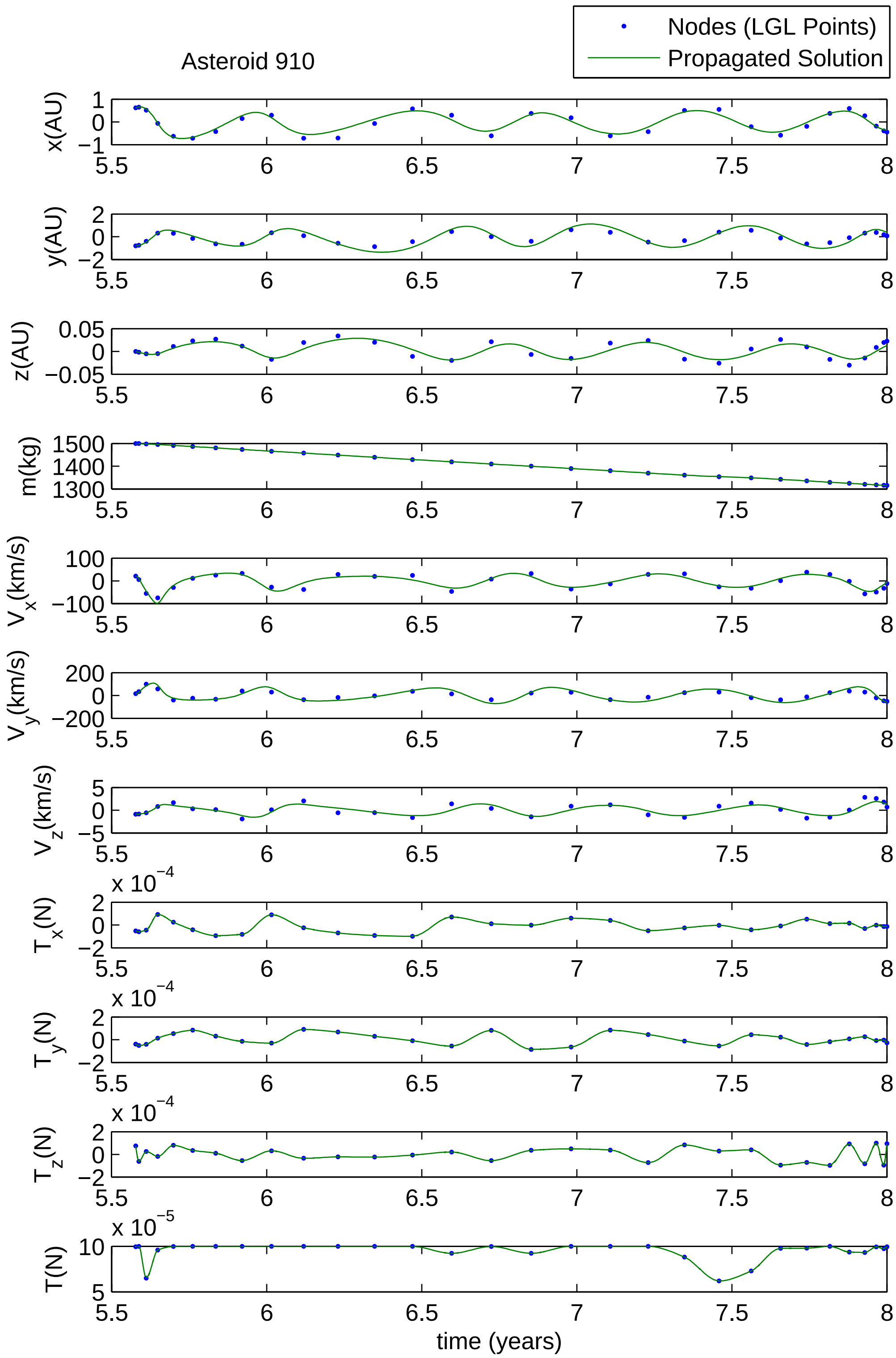

Figure 22: Rescaled 30 Node Visual Feasibility Plot for Asteroid 910

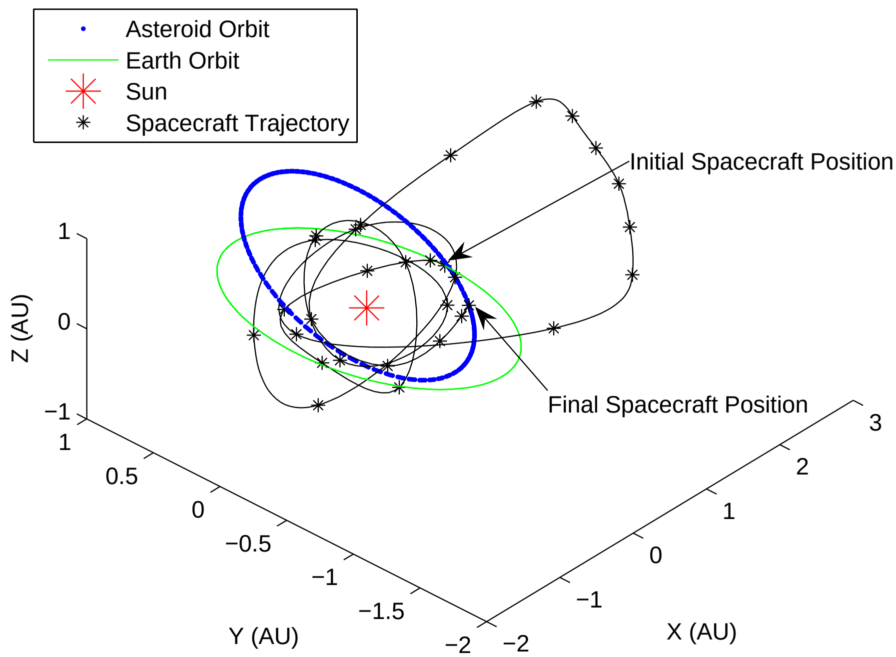

Figure 23: Rescaled 3-D Plot of Spacecraft Trajectory from Earth to Asteroid 906

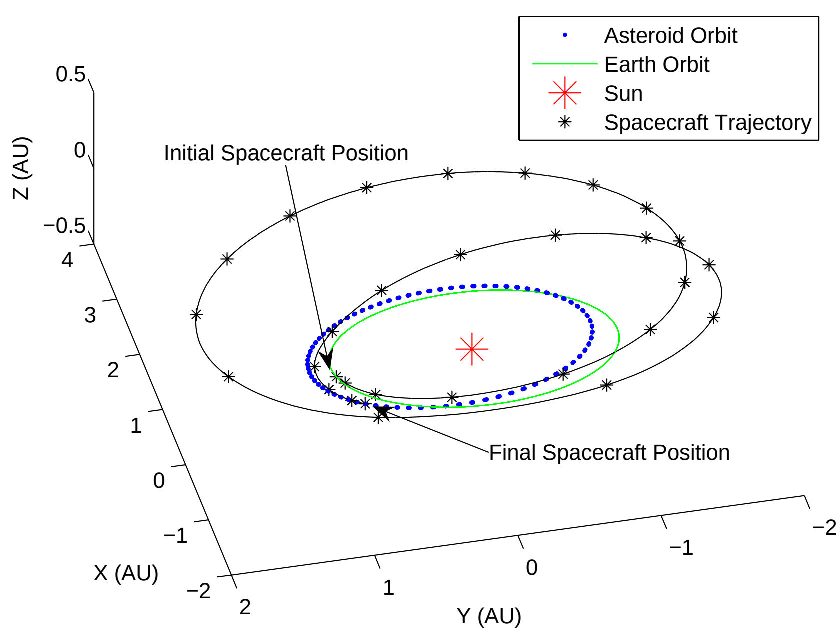

Figure 24: Rescaled 3-D Plot of Spacecraft Trajectory from Earth to Asteroid 907

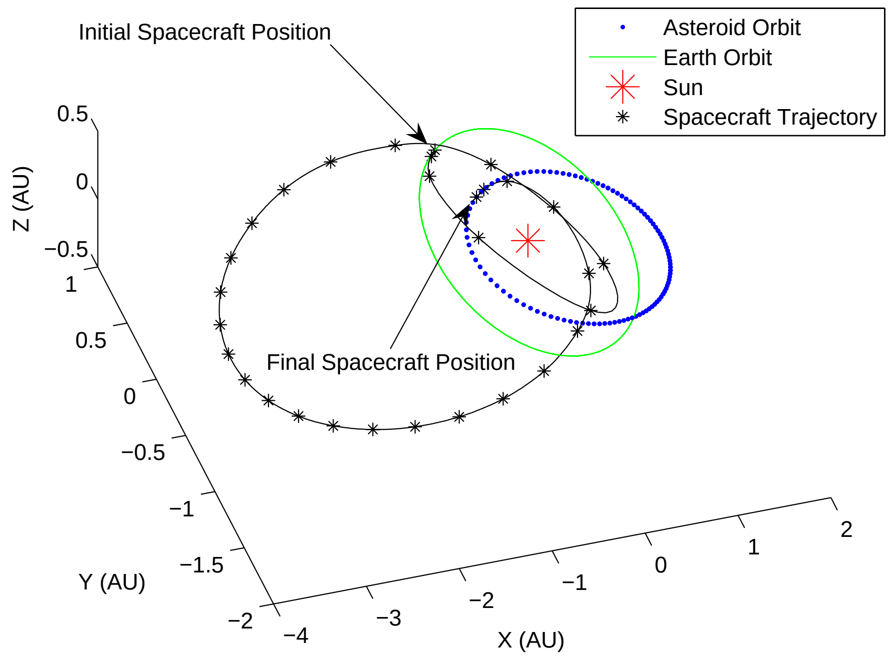

Figure 25: Rescaled 3-D Plot of Spacecraft Trajectory from Earth to Asteroid 908

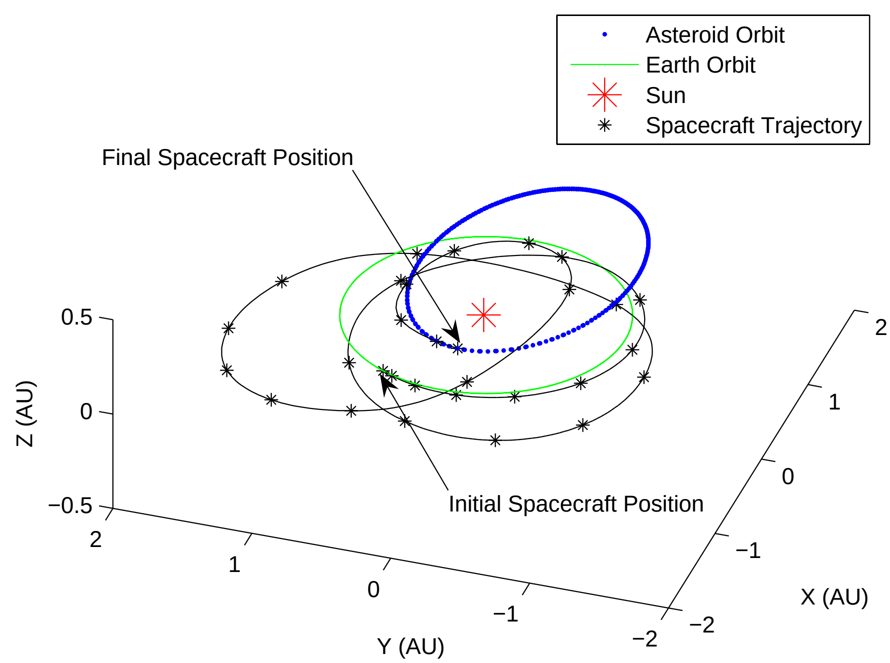

Figure 26: Rescaled 3-D Plot of Spacecraft Trajectory from Earth to Asteroid 909

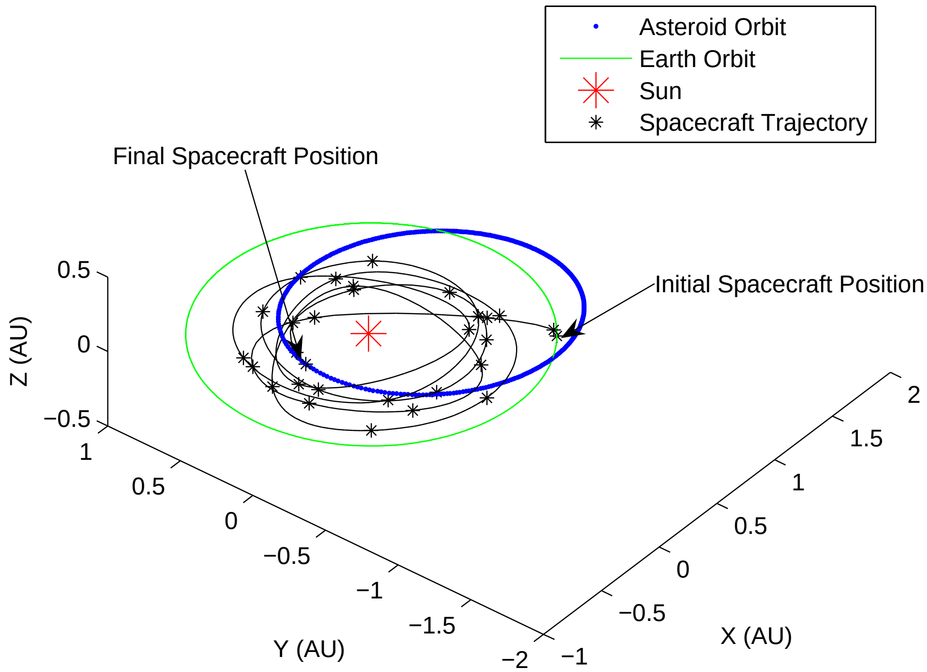

Figure 27: Rescaled 3-D Plot of Spacecraft Trajectory from Earth to Asteroid 910

Now that the model is producing feasible solutions, the work is focused on producing a selection routine that will select the optimal asteroid with which to rendezvous.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ASTEROID SELECTION

A. SELECTION LOGIC

With a base of knowledge for the expected solution, based on the data generated using the one-to-one Earth-asteroid rendezvous, the problem can now be formulated to include multiple asteroids. The desired outcome is to find one asteroid from the group with the least cost. The selection logic is implemented into the events file. A simple selection algorithm can be created that will automatically select an asteroid by finding the states of the asteroids at the final time, the final positions (x_f, y_f, z_f) and velocities ($v_{x_f}, v_{y_f}, v_{z_f}$) of the asteroids are known. Comparing these values to the final position and velocity of the spacecraft, the asteroid with the matching final state values can be determined and selected as the target asteroid.

$$\min\left[\left(v_{x_{S/C_f}}^2 - v_{x_{Ast_f}}^2\right) + \left(v_{y_{S/C_f}}^2 - v_{y_{Ast_f}}^2\right) + \left(v_{z_{S/C_f}}^2 - v_{z_{Ast_f}}^2\right)\right] \tag{3.8}$$

This is a very simple algorithm designed to fulfill the asteroid selection requirement of the problem.

B. SELECTION RESULTS

Using this selection method, DIDO is now run to find an optimal solution to any one of the five asteroids. Knowing the cost to go to each individual asteroid (Table 7), it is expected that DIDO will provide a control solution for a rendezvous with asteroid 910 based on the results of the one-to-one transfers. The results, displayed in Table 9, show that asteroid 910 was not selected.

Table 9: DIDO Output for a Model with Five Asteroids

Asteroid Chosen	Cost	Visually Feasible
907	0.34364	No

This is a very surprising result. The solution was found to be infeasible based on the visual check (see Figure 29). The cost of this solution is much greater than expected. Also, DIDO selected a different asteroid than expected. This solution is obviously not the optimal trajectory for an asteroid rendezvous. This selection technique did not produce a working solution. A new approach is, therefore, necessary.

The optimal trajectory is most likely the trajectory which requires the least Δv for the spacecraft, as it would require the least fuel and is closest to the Earth. The velocity of the asteroids at the final time can be compared to the velocity of the Earth at the initial time to determine the asteroid requiring the least Δv for rendezvous.

$$\min\left(\bar{v}_{Ast_f} - \bar{v}_{Earth_0}\right) \tag{3.9}$$

This is another technique for asteroid selection. This method of selection takes into account only the mass optimization facet of the problem at hand, however. DIDO is run again, implementing this new selection process; the expected selection is still Asteroid 910. This technique chose asteroid 907 again. The data is shown in Table 10. Unlike with the technique used earlier, however, this solution is feasible. The feasibility plot is shown in Figure 30 and the endpoint constraints in Table 11 and Figure 28.

Table 10: DIDO Output for a Model with Five Asteroids with new Selection Technique

Asteroid Chosen	Cost	Visually Feasible
907	0.24966	Yes

Table 11: Endpoint Constraints for new Selection Technique

Asteroid	Initial Postion	Initial Velocity	Spacecraft Mass	Final Position	Final Velocity	M_{Earth}	M_{Ast}
Lower Limit	0	0	0	0	0	0	0
Upper Limit	0	122.5	0	0.01	0.01	0	0
907	2.299E-7	122.5000	2.265E-14	0.01005	0.01007	1.20E-14	2.77E-13

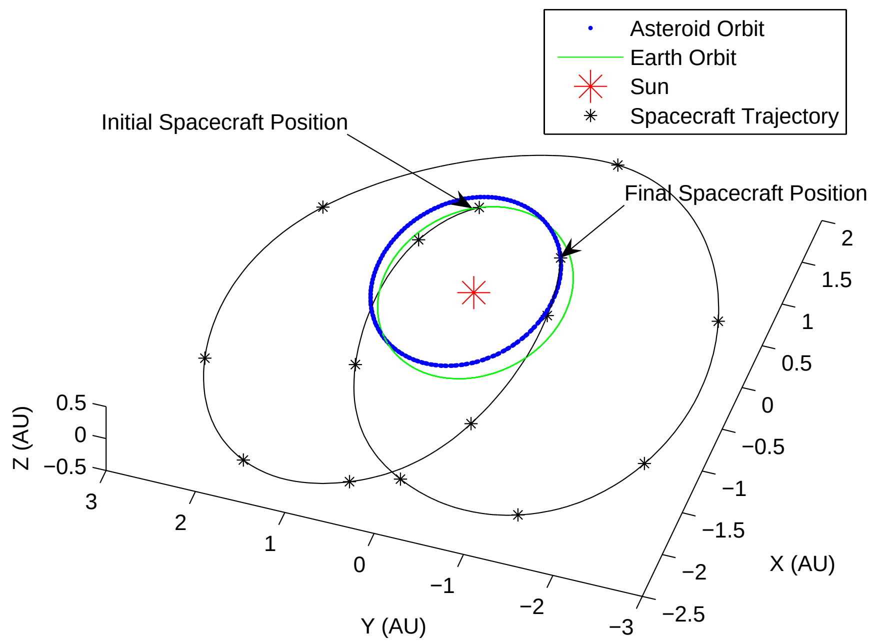

Figure 28: 3-D Plot of Spacecraft Trajectory with respect to the Earth and Asteroid 907 based on the New Selection Technique

While DIDO produced a feasible solution, the cost of the solution is much greater than desired. The one-to-one rendezvous with Asteroid 907 had the highest cost of all the one-to-one transfers. Either the selection criterion is flawed or DIDO has located a local minimum. In order to further understand this result, a discussion on global optimization is necessary.

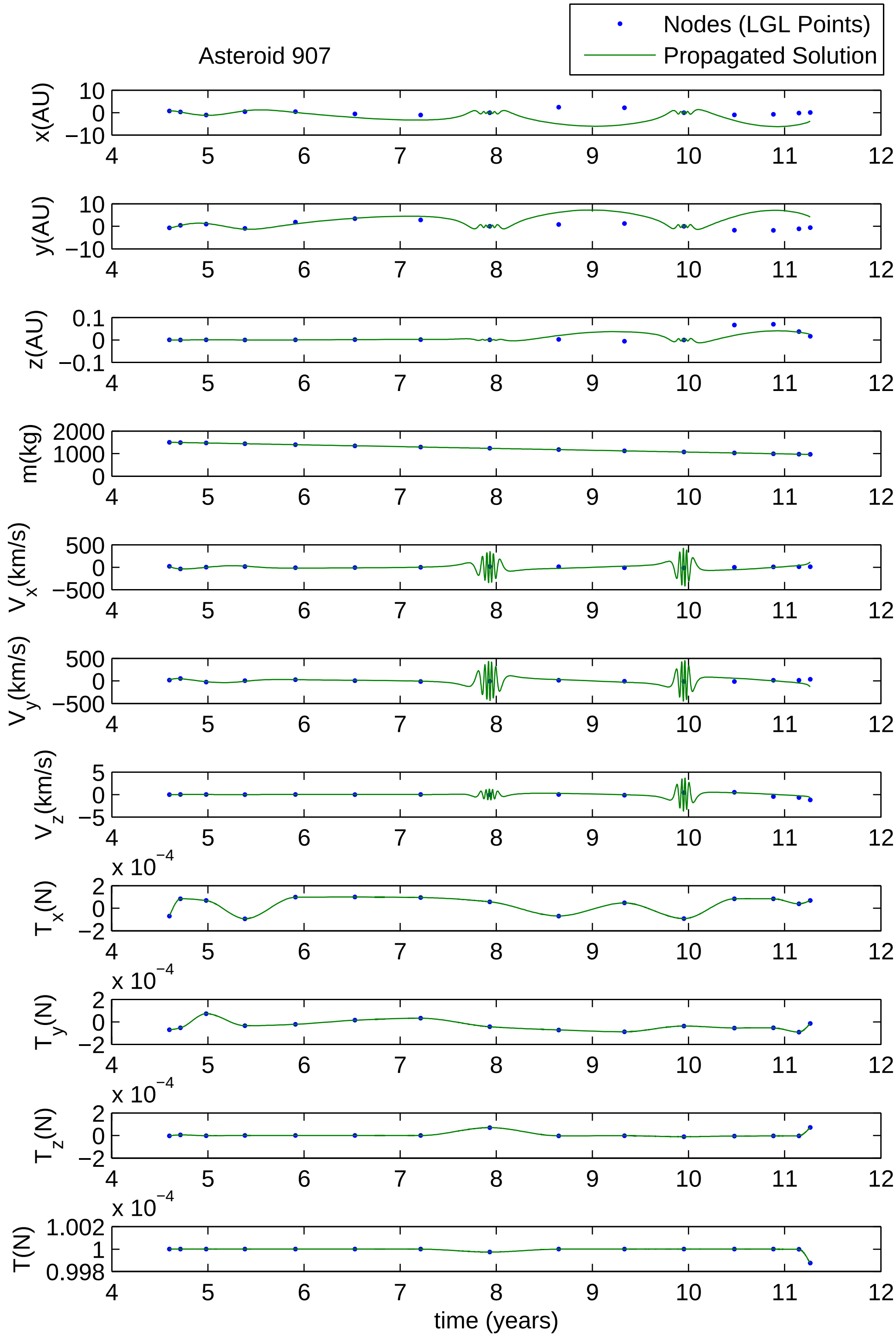

Figure 29: Visual Feasibility of Solution with Selection of Asteroid 907

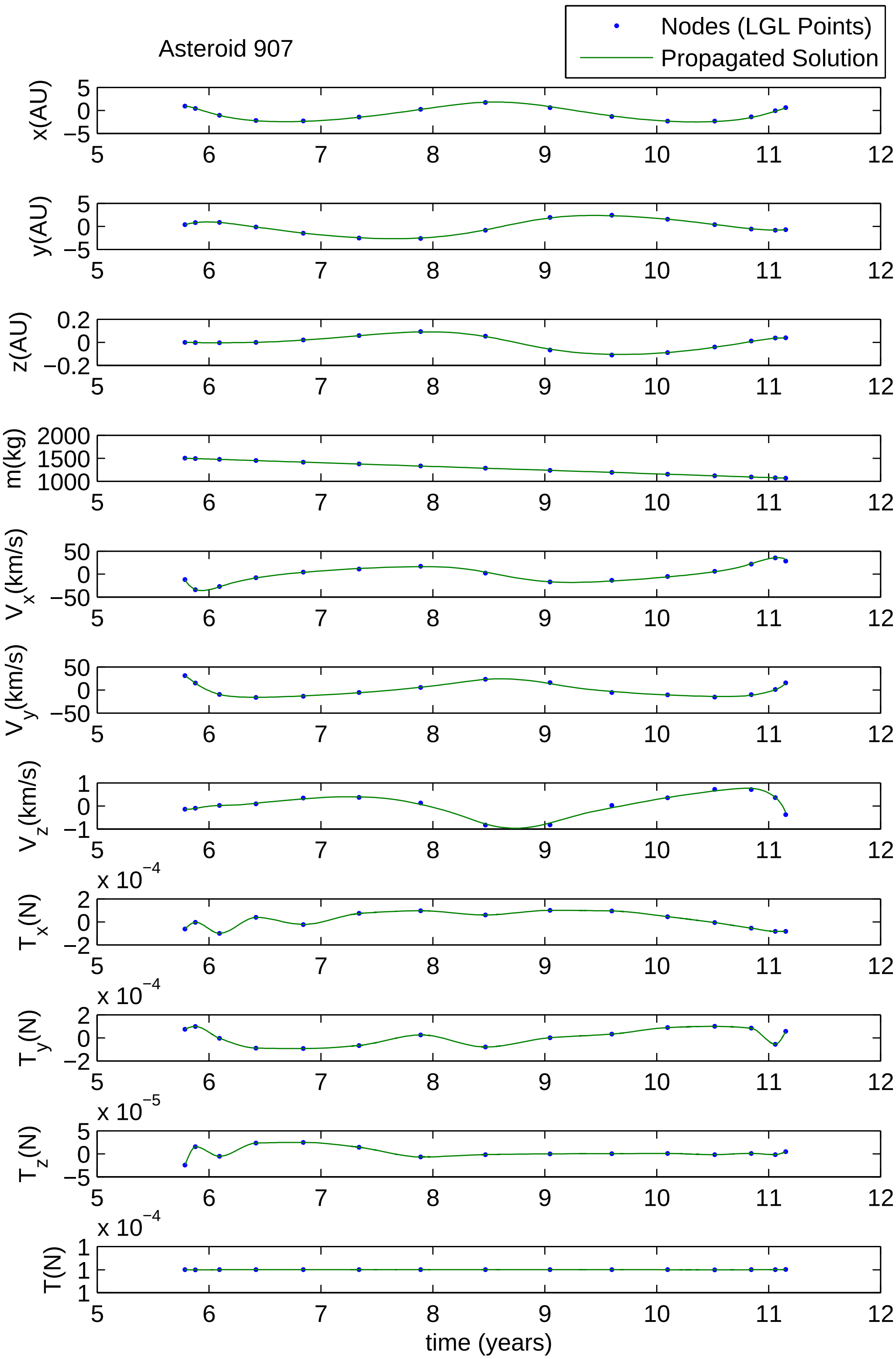

Figure 30: Feasibility of Selected Asteroid Utilizing New Selection Technique

C. GLOBAL OPTIMALITY

The purpose of this section is to familiarize the reader with the difference between a local and global optimum and how this was utilized in this work. A function may have many local minima; however, they can have only one globally minimum value. This concept is shown in Figure 31.

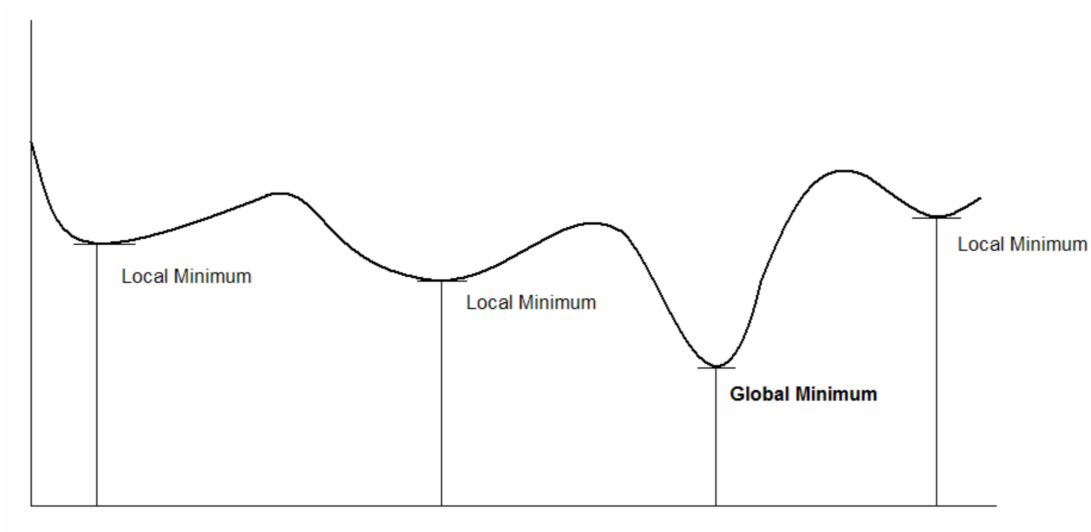

Figure 31: Demonstration of Global Minimum

It is clear that this illustrative function in Figure 31 has five maxima and four minima. Except for the end-points, these points are where the slope of the function is zero $\left(\frac{dy}{dx} = 0\right)$. The smallest of the minima is known as the global minimum, it is the smallest value which the function takes over the entire interval. It is this global minimum that we are searching for in this work.

1. DIDO and Global Minima

DIDO searches for an optimal solution. However, once it has finds a local extremal, it will stop its search and output the result. If the function under investigation has several local minima, as in the function shown in Figure 31, it is possible that DIDO would find only a local minimum. The version of DIDO used

in this work does not search for the global optimum. As a result, if the user wants to find a global optimum, it becomes necessary to find a technique that would lead to this value. This is done by repeatedly running DIDO while introducing another constraint: an upper bound on the final cost that is lower than that the previously calculated cost.

Earlier, while discussing the model used to find the solution of the problem, the events file was described as containing the specific boundary constraints of the problem. To help DIDO find the global optimum, and confirm global optimality, the final cost of the maneuver is added as an additional constraint and the problem is resolved. If we repeatedly constrain the values of the cost function to less that what was found in previous iterations until there is no longer any feasible solution the cost of the prior run was the global minimum and therefore the global optimal trajectory is determined. Using this technique, DIDO has to be run several times, once for each new decrease in the bound on the cost. This can become a time consuming practice, but its payoff is the acquisition of the globally optimal solution. This technique for searching for the global optimum will be called gradual cost-constrained optimization.

2. Gradual Cost-Constrained Optimization Results

The results found during the initial testing of the selection methods were discouraging. We can use the result of the selection as a jumping off point to start the search for the global minimum using gradual cost guiding. We know that there a minimum exists which is less than 0.25 based on the data collected during the one-to-one transfer study. By forcing the cost function to be less than 0.1245, DIDO's output must be a trajectory to asteroid 910 or a different asteroid with a lower cost than the one-to-one rendezvous. From this we should get a result to continue our search with. Once DIDO no longer finds a feasible result within the constraints, we know that we had attained the globally optimal trajectory in the previous run. This value should be equal to or less than the

minimum cost found using the one-to-one transfer performed earlier. The results from the tests using gradual cost-constrained technique are shown in Table 12.

Table 12: Selection Among Five Asteroids (906-910) with the Gradual Cost-Constrained Technique

Limit on Cost	Asteroid Chosen	Cost	Visually Feasible
No Limit	907	0.2499	Yes
0.125	909	0.0911	Yes
0.09	907	0.2618	No

It can be seen that from the initial starting point, the introduction of the upper bound greatly reduced the cost. The cost produced by this trial is slightly less than that generated by the one-to-one transfer to asteroid 910, the lowest cost previously found. When this cost was used as the upper bound, DIDO produced a solution with a cost greater than the bound, immediately nullifying the solution by being outside of the end-point constraints. The solution generated on the previous run is therefore suspected to be the optimal control because no solution could be found with a lower cost.

D. VALIDATION OF RESULT

Before concluding that the control solution produced in the previous section is a globally optimal trajectory it still must undergo a through verification process, the same way we did for the one-to-one rendezvous in Chapter III. The validity of the solution will be examined from several angles. First, the solution must meet all of the boundary conditions set up by the problem. In addition, the propagated control should show the feasibility of the results. It also must meet the necessary conditions for optimality, as developed in Chapter II.

Using the same techniques as before, the validity of this solution must also be checked for visual feasibility. The result generated earlier, in Section C.2, rendezvousing with asteroid 909 resulting in a cost of 0.08564, is shown to be visually feasible based on Figure 32. In addition, the boundary conditions must be met. These values, shown in Table 13, demonstrate that the end-point conditions are not met.

Table 13: Boundary Conditions for the Selected Trajectory Solution to Asteroid 909

Asteroid	Initial Postion	Initial Velocity	Spacecraft Mass	Final Position	Final Velocity	M_{Earth}	M_{Ast}
Lower Limit	0	0	0	0	0	0	0
Upper Limit	0	122.5	0	0.01	0.01	0	0
909	1.386E-4	122.1785	0	3.5556	0.1604	-4.8373E-9	-2.3444

The initial velocity, position, mass, and mean anomalies were within the bounds of the problem. The final position and velocity, on the other hand, were not within the necessary bounding values. This problem is not evident, however, in the plot of the trajectory of the flight with respect to the orbits of the earth and asteroid as the spacecraft ends on the asteroid's orbit. This plot is shown in Figure 33. It is evident from the values of Table 13 that the spacecraft does indeed start at earth, but it does not rendezvous with the asteroid as the final point on the trajectory does not lie on the orbit of the asteroid. This solution is, therefore, infeasible. Checking the solution against the necessary conditions is not necessary as the solution has been found to be infeasible. This selection technique has been proved to be inadequate as the optimal solution was not generated through its use.

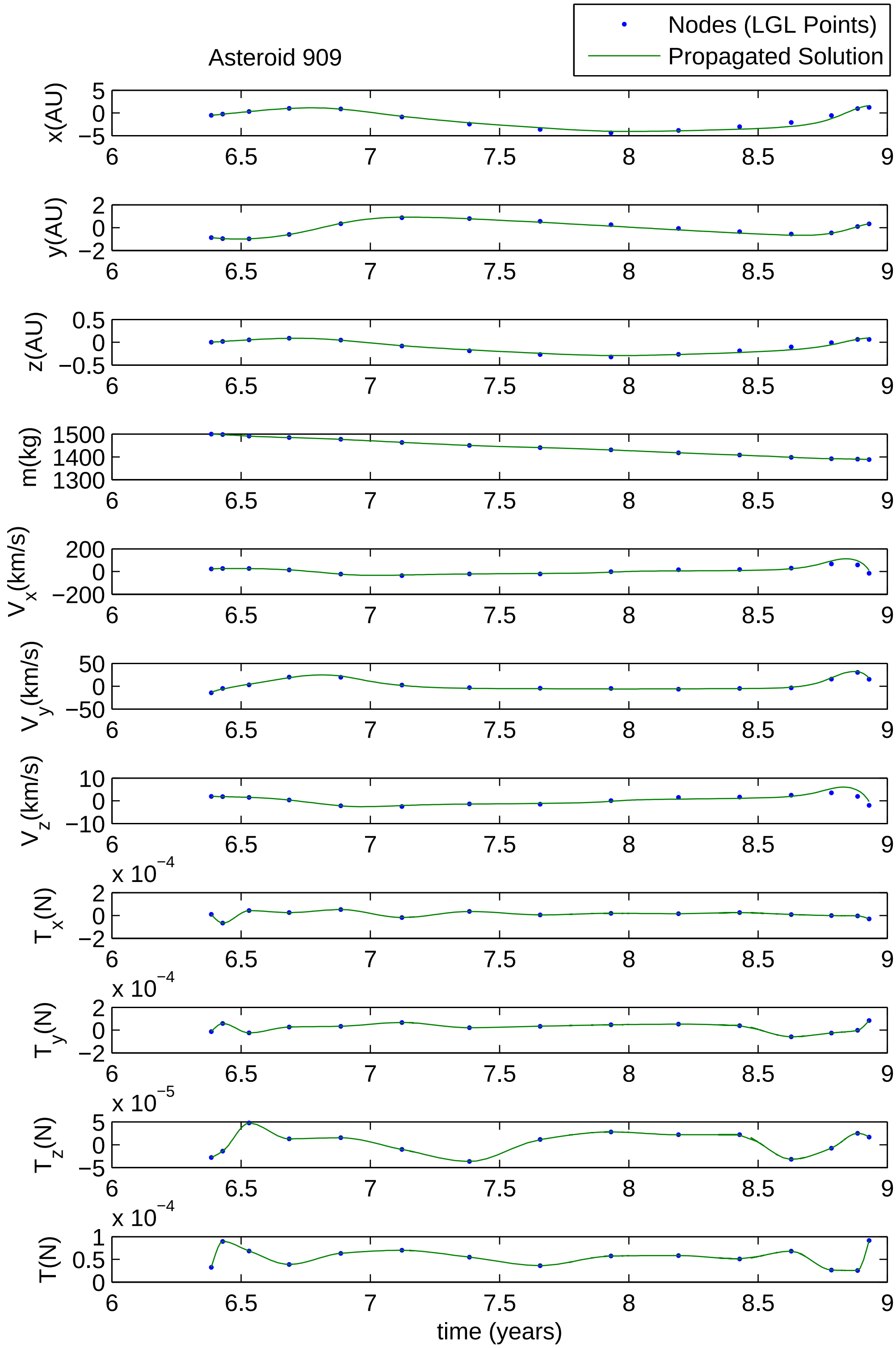

Figure 32: Visual Feasibility of Selected Trajectory Solution to Asteroid 909

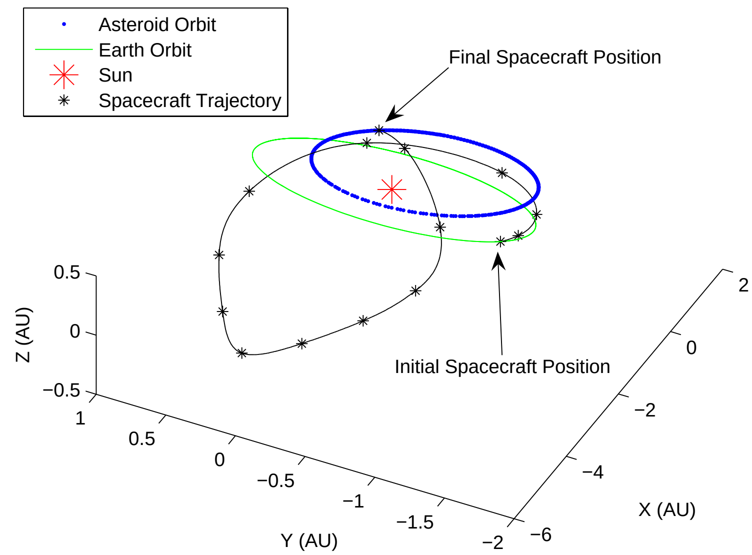

Figure 33: 3-D Plot of Spacecraft Trajectory with respect to the Orbits of the Earth and Target Asteroid 909

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS

A. INTRODUCTION

The purpose of this chapter is to give insight gained from working on this project as well as guidance for the continuation of work on this topic. There is a great deal yet to be done with this problem as it has not yet been fully solved. Hopefully the reader can gain some advantage when undertaking the continuation of this work based on what was learned while this research was being conducted.

B. SUGGESTIONS

The first and most important step for anybody trying to do work with DIDO is to familiarize themselves thoroughly with the way that it works. DIDO is a very powerful tool for finding optimal control solutions, if it is used properly. DIDO requires the problem to be formulated in a specific way in order to get the correct output. Having a strong understanding of the requirements and structure used by DIDO will relieve the researcher of a great deal of headache and time lost due to lack of understanding and errors. For more information regarding the use of DIDO, see the DIDO user's manual (Reference 8).

If not properly scaled, DIDO can take a very long time to run. Some of the trials completed for this work required over ten hours to run. These were run on a single PC without any special computing power. The runtime may be decreased if a more powerful computing source with several cores were used to run the iterations. Another way to decrease runtime is through improved scaling and elegant simplifications of the code. Regardless of manner in which it is accomplished, this work would be greatly simplified and improved if DIDO's runtime could be decreased.

Furthermore, a great deal can be learned from the oversight present in this work. The focus, here, was to find a selection process which would choose the optimal asteroid with which to rendezvous. The desire to find this value overpowered the logical progression of solving the problem. The trajectories created in the initial one-to-one Earth-asteroid rendezvous were taken as correct simply because the propagation of the control solution provided a visually feasible answer. As a result, the error in the model was not found until later. Consequentially, a great deal of time was lost once the error was found when checking the validity of the selected solution through checking the end-point constraints. Every solution must be checked for feasibility, constraints satisfaction, and necessary conditions for optimality before any assumptions of their optimality can be made. Any future work should be sure to correct this oversight present in this work.

C. FUTURE WORK

There is a great deal of work left to do on this project. Once the accuracy of the model is confirmed, the next steps can be taken. A selection process must be developed that can account for the complex system at hand. The problem has multiple layers in that the best asteroid must be selected and the best trajectory to that asteroid generated. The techniques used in this research to attempt asteroid selection failed to solve this complexity. The first technique used assumed linear distances could be used to select an asteroid. The bodies, however, move non-linearly as regulated by the equations of motion. The second technique, minimizing the Δv of the motion, failed to take the time optimization component of the problem into account. Other techniques need to be explored for the selection process.

Another source of more progress in this work is the propagation of the control. In practice, the only controls available in this problem are T_x, T_y, and T_z; therefore, $\bar{T}$, should be calculated directly from them. When propagating the solution, however, the $\bar{T}$ used is the value generated by DIDO. Although the

constraint on $\bar{T}^2 = T_x^2 + T_y^2 + T_z^2$ should be enforced when calculating T_x, T_y, and T_z, it is sometimes not quite as accurate as desired. More accurate knowledge of the feasibility of the solution could be gained by propagating the solution with $\bar{T}$ calculated from T_x, T_y and T_z instead of the $\bar{T}$ generated by DIDO.

It should also be noted that this work focused only on the first step of the GTOC2 problem. Once the spacecraft has rendezvoused with one asteroid in a group, GTOC2 requires it to rendezvous with three other asteroids in other groups. The selection of which group to visit first could be the next step taken once this problem has been sufficiently solved. Needless to say, a great deal of work still remains to be done in this problem and in the subsequent components of GTOC2. Trajectory optimization is an exciting and valuable field, the importance of which is growing exponentially. As technology advances, mankind's ability to improve upon the foundations of the past progresses as well.

D. CONCLUDING REMARKS

Creating a solution to the Global Trajectory Optimization Competition problem proposed by the Jet Propulsion Laboratory proved to be a challenging endeavor; one with a great deal of work remaining. It is always a frustrating situation when the limitations of time prevent the completion of a project as it has here. The foundation for the problem has been established, some tripping points have been identified, some obstacles overcome and still much remains to be accomplished.

The goal of this thesis was to develop an optimal trajectory for rendezvous with an asteroid in a group. The selection process still requires some attention as the optimal asteroid is not being selected based on the tests executed in this work. The greatest success of this research was the generation of one-to-one Earth-asteroid rendezvous trajectories. Without this data, the asteroid selection technique cannot be confirmed. The generation of these trajectories also demonstrates that the problem is properly formulated and that DIDO can produce

feasible solutions. The gradual cost-constrained optimization technique could be a valuable tool for anybody pursuing this problem. The use of this technique is not unique to this problem. Any optimal control problem that may encounter issues differentiating local and global minima could utilize the gradual cost-constrained technique to try to positively identify the global minimum. The gradual cost constrained technique can be used throughout the optimal control field. Any DIDO user can easily implement this useful tool to enhance their solution. Finding a global minimum is, by definition, the goal of every optimal control problem. In certain circumstances, this is a very useful method which may prove to be useful in the search for this illusive value.

APPENDIX A: ASTEROID DATA FILE

ID#	Semi-Major Axis (AU)	Eccentricity	Inclination (deg)	LAAN (deg)	Argument of Periapsis (deg)	Mean Anomaly (deg)	Epoch (JD)	Asteroid Cluster #
Group	1							
2011542	3.9501468	0.2391642	6.87574	16.88982	48.9603	229.49648	54000	1
2001038	3.9619932	0.227483	9.22988	58.20488	307.19412	201.38868	54000	1
2000624	5.2272807	0.023528317	18.193628	342.80533	185.13006	151.0004	54000	1
2016560	5.0765482	0.0408516	15.29915	100.75566	157.40487	261.64642	54000	1
2001902	3.971589	0.2236554	12.49318	59.4253	267.89541	47.99795	54000	1
2003552	4.2298497	0.71337656	30.907935	350.30192	316.99246	237.325	54000	1
2002456	5.1372368	0.074439	13.90266	327.41517	93.63287	230.61779	54000	1
2015278	3.9846457	0.2157149	9.29657	344.88781	64.13255	258.62785	54000	1
2001173	5.3227896	0.1371044	6.90993	283.91796	40.01722	205.92255	54000	1
3117599	3.0704328	0.5119076	21.62245	355.71658	225.09482	290.06416	54000	1
2001437	5.1600747	0.0433457	20.52178	315.83789	129.75256	244.24908	54000	1
2014569	3.977245	0.2859047	10.90696	346.1705	37.5375	338.80357	54000	1
2000588	5.1947906	0.1465778	10.32134	316.59634	132.48447	216.43354	54000	1
2002223	5.1985031	0.0140337	15.99057	220.98833	51.37102	279.17581	54000	1
2001754	3.9500648	0.1672029	12.11822	163.24455	111.93204	295.64241	54000	1
2012714	5.2062335	0.0356216	9.51657	298.91469	163.16174	210.69903	54000	1
2007119	5.2067455	0.1031099	19.26732	285.66811	119.08349	235.10541	54000	1
2014268	5.2873272	0.0906027	14.94484	284.67115	123.62387	237.9447	54000	1
2003134	3.9793318	0.2201713	7.63747	257.10507	163.00334	272.12822	54000	1
2000659	5.1922184	0.1166258	4.51942	350.88872	341.45996	318.17164	54000	1
2015436	5.2056216	0.0439938	16.26675	253.42915	178.28135	227.83254	54000	1
2001873	5.2491321	0.0922724	21.85938	197.91284	356.47324	335.83439	54000	1
2012444	5.2602165	0.0713189	30.80052	213.19945	64.17095	256.62017	54000	1
2000225	3.3888671	0.2668163	20.887897	197.19179	104.2599	310.40788	54000	1
3091801	7.2342917	0.6863551	24.22652	144.27642	240.32669	85.9533	54000	1
2001172	5.1915116	0.10319405	16.683194	247.39536	49.477769	250.4737	54000	1
2001208	5.2374808	0.0909268	33.56575	48.55332	295.69051	204.17234	54000	1
2001867	5.1330811	0.0437056	26.9088	283.69667	358.70429	250.58354	54000	1
2006984	3.9692255	0.1871492	16.80098	206.29149	247.90573	216.503	54000	1
2032511	5.0492697	0.4282629	8.93717	285.89591	345.49142	172.48229	54000	1
2002483	3.9660655	0.2763629	4.49881	252.1627	182.86999	226.77818	54000	1
2002920	5.1130804	0.0269571	21.11915	230.96163	196.35511	223.31577	54000	1
2001345	3.9798138	0.1808098	11.40178	137.5123	332.82006	181.8755	54000	1
2000911	5.25414	0.065868016	21.788152	338.02074	80.347917	253.5207	54000	1
3079876	3.1268516	0.5748146	20.68275	245.06986	115.06575	323.76911	54000	1
2003709	5.2609102	0.0614257	19.60369	187.17563	245.59639	219.05819	54000	1
2000361	3.9554449	0.21241778	12.631655	18.960258	68.162409	239.42006	54000	1
3046844	2.8266851	0.85854077	20.464243	326.68865	138.58443	74.835735	54000	1
2009799	5.1942001	0.0479145	30.51326	259.54994	113.20009	280.79516	54000	1
2002674	5.1717545	0.0678804	1.85443	179.86277	37.74182	321.3662	54000	1
2002893	5.17689	0.0764874	14.64719	108.76807	171.07065	241.30879	54000	1

2011396	5.2054308	0.0640564	12.58719	195.68829	175.96375	289.49289	54000	1
2000617	5.2266795	0.1381751	22.03587	44.35812	307.5839	183.22681	54000	1
2002241	5.2059683	0.0670944	16.60687	267.98621	291.17519	324.5333	54000	1
2003063	5.1854315	0.0584969	12.17397	287.88023	204.89888	178.85982	54000	1
2004834	5.2357128	0.1355501	28.45078	76.07509	350.67143	223.76284	54000	1
3169278	3.1343661	0.5204205	20.4243	252.16872	250.66259	169.20267	54000	1
2002959	3.9440841	0.2742882	5.23244	121.33837	284.68371	297.55373	54000	1
2020898	4.2287314	0.4647309	45.4942	293.29012	234.87212	212.42781	54000	1
2005209	5.1533221	0.0495159	9.06609	322.75689	104.50996	257.03537	54000	1
2002760	3.9834018	0.1223004	13.46141	352.816	155.94779	327.81918	54000	1
2004489	5.2761594	0.0602848	22.16378	86.70204	5.3875	190.45873	54000	1
2016070	5.1235811	0.1236231	16.25994	300.91274	352.61268	243.85689	54000	1
2002357	5.1921105	0.0438295	2.66967	179.31425	72.48815	292.0237	54000	1
2005264	5.2043321	0.1107648	33.57519	121.90801	358.9272	178.74752	54000	1
2002906	3.1616197	0.1137911	30.6904	84.60812	294.92209	165.70717	54000	1
2003708	5.220393	0.1577286	13.36738	291.17902	56.85664	189.19123	54000	1
2004833	5.2562716	0.0928909	34.68256	101.76168	278.74281	271.01003	54000	1
2004086	5.2256622	0.1209423	21.7387	54.95767	356.03329	270.21371	54000	1
2003548	5.1604558	0.0888397	8.07572	43.54071	26.8289	246.50665	54000	1
2006090	5.3136952	0.0571189	20.18055	328.52802	72.38497	259.09504	54000	1
3061681	3.346388	0.5398696	22.15021	277.03743	145.21595	341.7947	54000	1
2001749	5.1722902	0.1078736	6.08844	341.00747	111.19397	197.09829	54000	1
2002759	5.1700791	0.0660091	21.96815	171.23528	8.41746	126.14053	54000	1
2004754	5.1909721	0.0080923	12.34371	155.23002	213.07335	186.6018	54000	1
2005144	5.2263273	0.2720738	8.89879	322.85738	330.63494	227.70228	54000	1
2004791	5.1792451	0.0461581	25.94351	261.4468	165.16477	99.89186	54000	1
3089425	5.3324536	0.4300086	42.44729	213.05295	193.03549	136.63782	54000	1
2004543	5.0957247	0.0959612	14.73139	325.4165	84.13779	262.49894	54000	1
2002363	5.1655208	0.0356465	32.19648	211.83521	51.98333	290.96291	54000	1
2000944	5.7544224	0.66019798	42.532955	21.557042	56.72622	43.435582	54000	1
2005130	5.2456965	0.0099851	15.70813	242.53246	103.5806	169.16343	54000	1
3081550	3.0440863	0.5018147	16.79417	260.38958	38.40177	356.90507	54000	1
2004836	5.1823797	0.1077418	19.41902	82.04271	34.03318	195.75845	54000	1
2004063	5.1695887	0.1181985	18.94705	113.52578	317.49315	233.45724	54000	1
2002797	5.1057194	0.0878189	22.39234	69.9464	47.69826	175.56344	54000	1
2003317	5.2132675	0.1259699	27.88043	135.9195	149.13296	258.45877	54000	1
2011351	5.2491115	0.0638223	11.5758	251.09162	159.70997	276.73337	54000	1
2005012	5.2652415	0.0857824	4.99793	34.82759	333.0567	293.9424	54000	1
2001583	5.1071568	0.0522215	28.55192	221.3685	186.33737	260.04144	54000	1
2005025	5.2111349	0.0753205	11.01688	347.86522	72.09646	233.10794	54000	1
2002260	5.1896664	0.0439744	17.78242	86.57157	319.93149	252.91054	54000	1
2001404	5.3026206	0.1135323	18.00703	332.97289	59.33793	267.83412	54000	1
2001362	3.2218943	0.3698017	24.2039	121.39505	262.11257	18.88301	54000	1
2100004	2.6090888	0.69471344	16.289474	77.431206	11.737654	142.26727	54000	1
2011395	5.2073823	0.0672427	24.14544	213.23624	117.37947	328.72511	54000	1
2002207	5.1267457	0.0172127	6.81023	159.1796	300.66192	81.0411	54000	1
3035962	4.6900143	0.7036003	10.21541	183.5988	111.51786	202.03203	54000	1
2016974	5.2149294	0.0702616	15.01434	241.62526	134.19018	272.57798	54000	1
2004837	3.1977504	0.1314229	28.22335	327.27954	42.92711	296.90452	54000	1
2007641	5.2186575	0.0535495	34.68862	242.05333	227.73421	184.77791	54000	1

2003793	5.1948389	0.0891835	20.92981	200.52401	262.18506	206.49881	54000	1
2004035	5.2795635	0.0562528	12.134	233.73474	197.12702	232.16471	54000	1
2005259	5.1886873	0.0731641	15.93135	67.46291	199.37861	37.88812	54000	1
2004709	5.1962516	0.0204462	25.51675	253.23497	89.40599	213.85689	54000	1
2009661	3.9462035	0.2336894	12.9873	56.85707	288.56144	8.57451	54000	1
Group	2							
2000010	3.1366114	0.11799269	3.8423915	283.45768	313.00295	56.250366	54000	2
2000016	2.9197742	0.13948318	3.0956171	150.34418	227.86709	101.78424	54000	2
2000021	2.4350514	0.16374276	3.0644634	80.91394	250.10776	231.18992	54000	2
2000022	2.9091742	0.10280545	13.710708	66.236811	356.08934	3.206616	54000	2
2000024	3.1307722	0.13212508	0.7597936	36.007074	107.94133	257.16605	54000	2
2000031	3.1498399	0.22591347	26.31623	31.239635	62.002727	356.86228	54000	2
2000034	2.6851564	0.1087005	5.50356	184.534	330.0982	178.03603	54000	2
2000035	2.9897564	0.228466	7.93736	353.8186	213.97512	58.38133	54000	2
2000036	2.7470479	0.30349954	18.431224	358.47404	47.141652	25.346092	54000	2
2000038	2.7424397	0.15148566	6.9542317	295.91795	168.63723	86.025664	54000	2
2000041	2.7654349	0.27184993	15.764855	178.16309	46.221564	226.08638	54000	2
2000047	2.8776367	0.13528824	4.984635	3.2446812	314.60266	204.79556	54000	2
2000054	2.7122777	0.1964305	11.80389	313.45001	345.58093	81.759176	54000	2
2000055	2.7585286	0.14476162	7.1847686	10.539693	4.2608489	93.446378	54000	2
2000058	2.6998953	0.043183319	5.0578201	161.29511	34.44369	352.92475	54000	2
2000066	2.6451	0.1733656	3.04729	7.66789	43.73999	46.22178	54000	2
2000069	2.980066	0.16704444	8.5813425	185.1207	289.98554	137.4484	54000	2
2000070	2.6162373	0.1812037	11.58479	47.80504	255.87084	78.00096	54000	2
2000074	2.7784206	0.23987342	4.0751328	197.31394	174.52383	15.554851	54000	2
2000075	2.673238	0.3047786	5.00253	359.48157	339.5614	3.77453	54000	2
2000078	2.6201204	0.20736367	8.6876426	333.58458	151.41908	330.57112	54000	2
2000081	2.8537425	0.2108515	7.81207	1.50533	50.1751	129.18942	54000	2
2000086	3.1156178	0.20751135	4.8208178	86.451459	307.76761	247.06828	54000	2
2000090	3.1571193	0.15623753	2.2197774	70.234113	242.52717	58.626905	54000	2
2000095	3.067408	0.1489283	12.99847	243.14905	155.04622	308.59388	54000	2
2000097	2.668392	0.25700227	11.78307	159.77806	268.6762	351.69754	54000	2
2000099	2.6637091	0.19688701	13.858287	41.678878	196.02815	281.54555	54000	2
2000104	3.1549271	0.15268145	2.7914565	41.872068	30.942654	188.30844	54000	2
2000105	2.3739092	0.17644901	21.460972	188.35744	56.502696	7.3945065	54000	2
2000107	3.4773045	0.078537037	10.047151	173.13537	309.88106	346.53934	54000	2
2000110	2.7340757	0.0781141	5.97379	56.99462	281.76827	284.77871	54000	2
2000111	2.594318	0.10084142	4.923668	305.89396	165.92105	61.815975	54000	2
2000120	3.1157251	0.059447364	6.9548919	341.51217	231.93546	132.32475	54000	2
2000121	3.4573207	0.13841548	7.599757	73.217451	295.98151	233.08272	54000	2
2000125	2.7420783	0.08125	4.65594	169.16097	110.22377	246.10837	54000	2
2000128	2.7489736	0.12733828	6.2542741	76.459095	302.41385	159.53128	54000	2
2000129	2.8677223	0.21285798	12.218021	136.44012	108.17944	90.343092	54000	2
2000132	2.6093823	0.3881	25.05101	258.91641	254.37088	276.89429	54000	2
2000134	2.5632069	0.1166245	11.588804	346.2139	83.692584	86.432954	54000	2
2000135	2.4280968	0.20653464	2.3056068	343.89794	340.03522	220.0661	54000	2
2000136	2.2865589	0.0847066	9.56953	186.53627	132.61377	346.10322	54000	2
2000137	3.1184487	0.21988728	13.42254	202.45635	106.78731	297.02006	54000	2
2000143	2.7624553	0.0703414	11.46947	333.23845	250.94068	174.63933	54000	2

2000144	2.6544514	0.23556544	4.8084344	76.486474	293.65938	40.157765	54000	2
2000145	2.6731693	0.14450786	12.636769	77.453037	44.944313	140.69001	54000	2
2000146	2.7189094	0.064905901	13.074324	84.178218	143.44602	130.23606	54000	2
2000147	3.1342865	0.0340697	1.93499	248.73039	106.84911	167.74868	54000	2
2000156	2.7331251	0.22220581	9.7483653	242.18207	337.90684	196.55934	54000	2
2000159	3.1010391	0.11097779	6.1274027	134.33009	335.49307	282.20833	54000	2
2000161	2.3795391	0.1374982	9.05318	18.78424	294.39304	102.63249	54000	2
2000163	2.367393	0.1903735	4.8058965	160.36447	297.49222	26.693449	54000	2
2000168	3.3757402	0.0673035	4.63413	206.46806	168.09446	325.53015	54000	2
2000171	3.1334027	0.12927424	2.5460896	100.54761	58.061999	89.493077	54000	2
2000173	2.7416607	0.20834644	14.207656	148.35779	228.00507	117.48297	54000	2
2000175	3.1847599	0.2324518	3.21884	21.39951	321.25122	130.21293	54000	2
2000185	2.7394441	0.1271362	23.22034	153.94657	224.0974	264.43276	54000	2
2000187	2.7321898	0.23670797	10.597906	21.904126	195.14152	153.41892	54000	2
2000194	2.6181429	0.23614207	18.485624	159.51944	162.83374	284.62328	54000	2
2000195	2.8803618	0.04057	6.96866	7.20269	123.4987	102.20799	54000	2
2000200	2.7369494	0.13381634	6.9014922	324.69997	86.014728	305.98131	54000	2
2000201	2.6787336	0.1792631	5.75756	157.11692	181.24502	349.42127	54000	2
2000205	2.7767208	0.0361837	10.69477	212.0672	171.02616	158.14767	54000	2
2000206	2.7403923	0.040846283	3.7803055	145.27928	302.04137	284.70264	54000	2
2000207	2.2836787	0.0286959	3.80295	29.28743	192.12819	87.89963	54000	2
2000209	3.1445726	0.062655847	7.1714093	0.78538285	250.07104	34.32137	54000	2
2000211	3.0409546	0.16227833	3.8821037	263.75641	174.76297	309.57543	54000	2
2000216	2.797122	0.25036608	13.133755	215.66472	179.3494	181.60763	54000	2
2000224	2.6444588	0.0460098	5.83838	353.0192	284.05593	257.00632	54000	2
2000232	2.5502931	0.1778829	6.0708	152.52032	51.12076	64.32308	54000	2
2000238	2.9086285	0.0879187	12.4025	184.19336	207.19158	178.19462	54000	2
2000240	2.6643756	0.2068432	2.10451	115.22263	300.50223	324.5337	54000	2
2000250	3.153134	0.1270277	12.82149	24.05332	75.74685	142.31316	54000	2
2000266	2.8043578	0.15736045	13.390532	236.00128	151.25527	16.632347	54000	2
2000304	2.4030494	0.2214414	15.83975	159.22264	172.3242	191.50577	54000	2
2000313	2.3756945	0.1794471	11.6459	176.82084	315.81027	13.19227	54000	2
2000325	3.202845	0.1680242	8.54318	345.28451	67.86966	111.00173	54000	2
2000326	2.3176268	0.19054712	23.724144	32.334512	238.50135	100.32007	54000	2
2000329	2.4764225	0.0238695	15.88431	178.54468	52.19986	96.05795	54000	2
2000334	3.885343	0.024215343	4.642853	130.2253	151.07984	5.201958	54000	2
2000338	2.9120186	0.0201877	6.03807	287.63451	123.54059	295.61868	54000	2
2000342	2.5681538	0.128678	7.34603	232.74759	225.4028	193.24988	54000	2
2000344	2.5948773	0.31596263	18.356075	48.25056	237.38162	172.27458	54000	2
2000345	2.3253732	0.061343146	9.751305	212.79371	229.68958	8.5651683	54000	2
2000347	2.6131906	0.1641544	11.69434	85.84551	84.86164	248.97015	54000	2
2000350	3.1112374	0.1561224	24.90012	90.19672	338.9865	249.8531	54000	2
2000356	2.7559652	0.2397877	8.23048	354.86413	78.80956	333.70592	54000	2
2000360	2.9986233	0.1817627	11.71228	132.65874	288.55851	27.11377	54000	2
2000369	2.6491767	0.0976895	12.7077	94.38445	269.55336	10.20211	54000	2
2000373	3.1131734	0.1465925	15.44736	4.0578	347.77644	202.41227	54000	2
2000375	3.1231061	0.107686	15.93372	336.67192	344.59878	192.61115	54000	2
2000380	2.6788222	0.1132572	6.15571	95.21741	240.54282	21.26849	54000	2
2000381	3.2204713	0.0955587	12.52422	125.34741	137.52796	102.99105	54000	2
2000382	3.1156156	0.1770362	7.40193	313.60394	270.78992	81.18543	54000	2

2000386	2.8945485	0.17292904	20.254208	166.94275	220.14756	65.82279	54000		2
2000388	3.0067013	0.05887506	6.4580588	354.62692	333.05707	49.966331	54000		2
2000393	2.7790703	0.33144266	14.870607	212.51713	91.047917	126.41939	54000		2
2000395	2.7858859	0.0838058	3.35143	259.62654	12.95644	167.79539	54000		2
2000404	2.5943355	0.1983143	14.115702	92.674305	121.11608	271.32457	54000		2
2000405	2.5837501	0.24463276	11.950979	255.29974	309.27005	353.02963	54000		2
2000407	2.6248052	0.0707861	7.53482	294.83119	82.01003	40.65893	54000		2
2000410	2.7285227	0.2365507	10.92226	97.21109	172.00263	59.08877	54000		2
2000413	2.584411	0.3431661	18.71569	103.92107	252.68584	308.8369	54000		2
2000414	3.5083344	0.0689192	9.54215	110.74346	326.64868	352.52959	54000		2
2000418	2.5920701	0.119635	6.82571	249.0928	125.96347	103.5344	54000		2
2000423	3.0676241	0.041046488	11.240443	69.554082	206.55983	290.60245	54000		2
2000429	2.6070974	0.12333371	9.5273188	220.04862	168.7474	325.59031	54000		2
2000441	2.8060136	0.0827663	8.1419	253.86288	201.35201	313.76537	54000		2
2000442	2.3449721	0.0714341	6.06243	135.03435	85.04871	278.05265	54000		2
2000444	2.771173	0.17286275	10.278705	195.8366	155.21028	322.30187	54000		2
2000445	3.1997043	0.1909852	21.37172	292.4142	79.8239	291.93822	54000		2
2000448	3.1381577	0.1843979	12.71486	37.37604	295.24661	107.79207	54000		2
2000449	2.5526781	0.1708912	3.08986	86.03696	46.37252	342.694	54000		2
2000466	3.3583767	0.082421992	19.16314	291.19859	245.93322	345.95823	54000		2
2000481	2.7389796	0.1582352	9.8583	67.0244	348.66307	80.4701	54000		2
2000488	3.1579038	0.1692412	11.500393	84.980185	68.917033	289.65081	54000		2
2000489	3.1535619	0.038868229	12.976896	167.18408	12.619788	7.9310261	54000		2
2000490	3.1685051	0.0989462	9.26516	178.5047	196.80207	120.23447	54000		2
2000494	2.9888645	0.0568189	7.07552	38.39191	216.78947	177.45085	54000		2
2000497	2.8579034	0.2973859	4.82186	6.77463	2.80643	228.05837	54000		2
2000498	2.6503781	0.2251495	9.50401	97.49661	241.2688	75.92796	54000		2
2000508	3.1607974	0.014003443	13.36355	44.510226	179.85303	129.14648	54000		2
2000511	3.1659016	0.18564446	15.938414	107.6717	338.52068	247.71727	54000		2
2000516	2.6796943	0.2735302	12.95664	328.88736	258.53574	296.93286	54000		2
2000521	2.7417331	0.2813483	10.59138	89.69902	316.09633	246.32803	54000		2
2000535	2.5690206	0.0229292	6.78274	84.87326	68.24631	20.05767	54000		2
2000558	2.9064565	0.0431148	8.3662	143.87801	317.36518	215.4325	54000		2
2000559	2.7122622	0.064323766	9.3105401	112.21021	130.74	212.48385	54000		2
2000566	3.3823202	0.1108082	4.89863	80.26508	294.61109	331.04968	54000		2
2000569	2.6561647	0.1828515	1.29584	301.97392	141.80532	34.01162	54000		2
2000583	3.1705289	0.1619588	8.25067	257.98239	253.69818	179.04336	54000		2
2000585	2.4303409	0.1296983	7.5572	180.37753	327.97351	205.455	54000		2
2000593	2.6975729	0.2179818	16.89206	76.18743	30.92169	286.61962	54000		2
2000602	3.0914405	0.2440861	15.07537	331.64478	45.83763	302.7661	54000		2
2000618	3.1900659	0.078422036	17.012019	111.19981	227.86543	239.39252	54000		2
2000635	3.1443395	0.0786644	11.03908	183.46648	219.42795	168.40125	54000		2
2000654	2.2968024	0.23192318	18.124633	278.57048	214.07158	112.35919	54000		2
2000701	3.0157244	0.0315655	7.1136	244.14352	313.54037	228.69453	54000		2
2000702	3.1943626	0.022868874	20.60234	289.97403	352.37054	337.72603	54000		2
2000712	2.5738464	0.1881205	12.781487	231.04829	181.1623	80.890211	54000		2
2000713	3.391033	0.16608003	10.359739	217.79282	136.08384	286.71598	54000		2
2000735	2.7292584	0.3214867	16.87956	43.03	310.136	332.75553	54000		2
2000751	2.5502414	0.153174	15.61479	78.93492	302.31969	163.86706	54000		2
2000755	3.1724209	0.1466565	3.23894	177.2636	42.40449	145.37775	54000		2

2000764	3.1850605	0.1070606	10.07278	259.37313	154.7591	100.1314	54000	2
2000772	3.0035141	0.092042448	28.783444	64.045057	142.09532	205.57057	54000	2
2000776	2.9350442	0.16086041	18.245702	79.867084	306.61868	249.80628	54000	2
2000785	2.5692316	0.2105935	12.73156	72.18331	129.96803	128.71164	54000	2
2000786	3.1687258	0.1671064	14.55243	89.92703	133.60068	115.60687	54000	2
2000791	3.115249	0.1996704	16.38494	130.11717	202.04482	217.95885	54000	2
2000798	3.0142313	0.0413488	9.22952	214.52891	41.83994	8.942	54000	2
2000814	3.1498691	0.3093096	21.83526	88.86011	297.08583	85.74032	54000	2
2000821	2.7776856	0.207103	5.37832	209.91865	32.90585	163.61022	54000	2
2000849	3.1547846	0.1953547	19.48619	228.51247	63.9943	160.73835	54000	2
2000860	2.7945762	0.1091712	13.31523	309.55072	19.97543	210.65456	54000	2
2000872	2.7313074	0.0784015	7.36689	194.94024	20.02667	299.6105	54000	2
2000907	2.7977888	0.1637967	19.57368	43.17347	88.29047	211.84409	54000	2
2000931	3.185335	0.2233883	11.44311	111.48085	313.89267	191.3093	54000	2
2000977	3.1153896	0.0293926	15.19978	75.90784	83.30353	169.14985	54000	2
2001015	3.208478	0.081469182	9.4580072	120.49107	282.54502	235.74511	54000	2
2001028	3.3947356	0.1185204	9.39108	63.48771	25.06466	47.074	54000	2
2001061	3.1383114	0.2082506	2.49726	91.29504	303.85151	225.10094	54000	2
2001082	3.1216608	0.1808789	1.85016	148.02332	188.61985	134.3123	54000	2
2001093	3.1304429	0.2707514	25.20975	55.69696	251.94642	183.03763	54000	2
2001102	3.0687473	0.1173315	15.81039	216.84653	114.79816	256.27357	54000	2
2001277	2.6992491	0.239092	6.9664	247.25447	47.368	145.46956	54000	2
2001445	3.1234889	0.1772867	2.28497	89.30058	270.99423	282.75712	54000	2
2001461	3.1272276	0.0423866	15.32426	104.84379	334.56144	191.21196	54000	2
2001580	2.1968759	0.48791122	52.097105	62.32879	159.50668	121.45456	54000	2
2001606	2.6913513	0.3150995	7.70009	190.77887	142.30677	251.45853	54000	2
2001625	3.2016194	0.2218822	15.55532	322.144	282.75245	181.27996	54000	2
2001639	2.5719099	0.1512801	8.42685	324.37971	105.46543	72.49232	54000	2
2001794	3.1288813	0.1552413	14.50739	221.50437	335.55787	231.28079	54000	2
2001931	2.5404735	0.27278	8.2461	182.53508	163.92287	45.11593	54000	2
2001963	2.4215853	0.21073058	25.051232	106.98322	355.82804	179.40627	54000	2
2002379	3.1646802	0.2762638	0.46741	151.27404	177.62201	154.27748	54000	2
2002407	2.9224974	0.2211028	2.47635	342.34897	10.80364	43.83693	54000	2
Group	3							
2000003	2.6676188	0.25819419	12.971682	170.12215	247.82331	75.986335	54000	3
2000005	2.5736689	0.19269048	5.3685634	141.6853	357.50991	266.07961	54000	3
2000006	2.4251131	0.20172572	14.752027	138.74385	239.55732	326.18977	54000	3
2000007	2.3854903	0.23142212	5.527283	259.72283	145.41092	349.83031	54000	3
2000008	2.2015458	0.15620731	5.8884175	110.96399	285.39792	246.60931	54000	3
2000009	2.3871957	0.1214414	5.5765165	68.973442	5.6901086	127.8487	54000	3
2000011	2.4522002	0.10011807	4.6247051	125.62803	195.29555	178.97445	54000	3
2000012	2.334669	0.22056537	8.3638054	235.53818	69.58648	301.90748	54000	3
2000014	2.5849886	0.16811947	9.1072838	86.461325	96.32871	156.36084	54000	3
2000015	2.643284	0.18718147	11.738272	293.27326	97.914615	309.03614	54000	3
2000017	2.4700724	0.1344116	5.58717	125.60807	136.00231	216.015	54000	3
2000018	2.2955326	0.21870798	10.125237	150.53455	228.00031	43.626263	54000	3
2000020	2.4091511	0.14287954	0.70691821	206.50811	255.50551	346.24964	54000	3
2000023	2.6273733	0.2329636	10.145257	67.227831	59.312781	351.82574	54000	3
2000025	2.3999647	0.25544399	21.584123	214.268	90.161994	33.436221	54000	3

2000026	2.656335	0.086901823	3.5621827	45.885083	193.16118	138.34082	54000	3
2000027	2.3476948	0.17191778	1.5837422	94.806111	356.77913	153.80173	54000	3
2000028	2.7780323	0.14824802	9.4013833	144.50295	342.55305	15.27945	54000	3
2000029	2.5540421	0.072585419	6.096443	356.49859	63.459167	253.78959	54000	3
2000030	2.3665367	0.12638171	2.0974832	307.77461	86.72257	223.47454	54000	3
2000032	2.5879007	0.08299776	5.5306247	220.57549	339.79731	22.315567	54000	3
2000033	2.865151	0.3376296	1.87042	8.59314	338.24659	128.61147	54000	3
2000037	2.6414184	0.17665525	3.0732741	7.4126651	62.695182	6.3427201	54000	3
2000039	2.7686996	0.11419824	10.382913	157.17103	209.57309	36.855297	54000	3
2000040	2.2678373	0.046566607	4.2556875	94.292217	268.90779	220.356	54000	3
2000042	2.4419555	0.22279671	8.5295276	84.397918	236.6316	96.043894	54000	3
2000043	2.2032648	0.16794702	3.4679469	264.93471	15.950712	191.974	54000	3
2000057	3.1493067	0.1182737	15.20008	199.33961	212.88978	50.32014	54000	3
2000060	2.393591	0.18203332	3.6020835	191.80383	270.41631	64.514348	54000	3
2000061	2.9819926	0.1678048	18.21925	333.7717	13.81601	110.25816	54000	3
2000063	2.395578	0.12605849	5.7855725	337.91498	295.63623	356.80652	54000	3
2000067	2.4213289	0.18484185	6.0269995	202.72444	106.29921	156.01781	54000	3
2000068	2.782613	0.18536679	7.9716224	44.182403	305.38918	3.378668	54000	3
2000071	2.7549414	0.176456	23.255563	316.10551	267.4553	352.31621	54000	3
2000073	2.6660162	0.041362	2.37313	7.23877	54.66187	191.9127	54000	3
2000079	2.4444158	0.19213223	4.6226653	206.80097	200.35481	123.7389	54000	3
2000080	2.296534	0.2003401	8.66477	218.82041	139.11469	234.60882	54000	3
2000082	2.7598915	0.2244582	2.8333141	25.636344	110.38231	269.9886	54000	3
2000089	2.5500653	0.18377079	16.140829	311.64793	45.00997	104.93539	54000	3
2000100	3.0933126	0.1650934	6.43021	127.33508	185.87533	130.37461	54000	3
2000101	2.5829817	0.14151875	10.198982	343.4749	347.82922	101.84486	54000	3
2000103	2.7026333	0.0795678	5.421	136.27878	190.13768	52.67649	54000	3
2000113	2.3755464	0.087703611	5.0372639	123.59522	79.053555	337.73369	54000	3
2000115	2.3806207	0.1915	11.59731	308.99545	96.7467	221.98288	54000	3
2000116	2.7695069	0.13755329	3.5690216	64.03691	93.101434	318.91078	54000	3
2000118	2.4371741	0.1633635	7.74344	47.7451	33.63437	198.28777	54000	3
2000119	2.5812942	0.0810231	5.77832	203.73791	171.29982	338.06631	54000	3
2000123	2.6942954	0.1218106	6.42781	307.95504	124.95905	16.10407	54000	3
2000124	2.6301198	0.076540038	2.9507408	188.18585	63.155748	230.10546	54000	3
2000126	2.4389509	0.105986	2.9245	23.47891	327.99036	91.12117	54000	3
2000138	2.4486111	0.1624635	3.20806	54.95009	260.0207	143.2047	54000	3
2000149	2.1744038	0.0653143	0.93695	159.64764	251.12748	253.78577	54000	3
2000151	2.5917414	0.0331371	6.4444	39.04645	134.55059	7.94691	54000	3
2000158	2.8685446	0.0565242	1.00337	278.55081	143.9388	3.12319	54000	3
2000167	2.8527952	0.0336352	2.21049	166.44873	125.83834	28.77731	54000	3
2000169	2.3578594	0.13075	5.50226	354.82138	334.69519	350.02999	54000	3
2000170	2.5532105	0.0646822	14.4023	301.46438	157.69423	300.03171	54000	3
2000172	2.3803732	0.1142941	10.03136	332.0841	359.24594	288.80348	54000	3
2000174	2.859052	0.1458847	12.12754	327.80693	289.76608	316.3647	54000	3
2000178	2.4599218	0.0438536	1.89948	51.19998	211.55291	281.47495	54000	3
2000179	2.970494	0.1155056	7.81766	252.12535	105.37094	104.12798	54000	3
2000180	2.7200002	0.1690067	0.87057	312.70655	175.49316	231.86403	54000	3
2000181	3.1402839	0.1982904	18.7985	143.59272	317.42132	113.3454	54000	3
2000182	2.4179605	0.18506679	2.0028901	107.27931	309.90481	156.27731	54000	3
2000183	2.7931846	0.349925	26.37466	142.01678	264.16968	253.8036	54000	3

2000186	2.3617979	0.1499119	13.17267	14.87064	315.24035	18.13794	54000	3
2000188	2.7629704	0.1775287	11.73414	241.2132	68.31891	295.83763	54000	3
2000189	2.4499896	0.0370351	5.17924	203.61856	166.82805	133.14401	54000	3
2000192	2.404096	0.24621664	6.8170145	343.41423	29.840171	221.89748	54000	3
2000196	3.1149136	0.02269419	7.2609296	72.55494	199.92548	346.38828	54000	3
2000197	2.7408956	0.1602645	8.79318	81.68189	246.00807	318.73523	54000	3
2000198	2.4593202	0.2278889	9.30913	268.52803	88.55903	290.62198	54000	3
2000202	3.0764769	0.0962809	8.82984	137.05498	0.94421	72.40597	54000	3
2000204	2.67313	0.1719995	8.27148	205.21481	55.82695	137.45271	54000	3
2000208	2.891708	0.0154312	1.74941	4.5539	125.39114	191.44426	54000	3
2000215	2.7669402	0.0344195	1.69006	25.05574	321.44768	34.44622	54000	3
2000218	2.6657744	0.11716641	15.226448	170.87994	60.735184	77.80577	54000	3
2000219	2.3539776	0.2230137	10.84215	200.9526	142.27416	336.21897	54000	3
2000221	3.0115033	0.1032757	10.8869	141.94429	195.91626	104.86267	54000	3
2000228	2.2013308	0.241223	2.53831	313.42898	18.77463	353.21101	54000	3
2000230	2.3832188	0.060890097	9.4382745	239.96029	139.39921	230.91656	54000	3
2000234	2.3858256	0.2442783	15.35266	144.63948	192.16925	134.46103	54000	3
2000235	2.8829304	0.0598554	9.02719	66.23928	211.97564	170.0847	54000	3
2000236	2.8031571	0.1873378	7.68294	186.13681	174.03894	258.17166	54000	3
2000237	2.7622571	0.0733517	9.75594	84.44051	201.72651	197.94377	54000	3
2000243	2.8611801	0.045985314	1.1375042	324.18546	108.3985	28.35885	54000	3
2000245	3.1015974	0.19769977	5.1773835	61.525116	327.52007	193.52253	54000	3
2000254	2.195187	0.1215113	4.51431	28.54322	233.09692	329.78941	54000	3
2000258	2.6152135	0.2051338	14.29304	207.70149	154.94955	38.8324	54000	3
2000262	2.5523258	0.2141044	7.70896	38.71081	24.60432	119.04846	54000	3
2000264	2.802004	0.1338519	10.43526	49.77904	340.02132	217.85622	54000	3
2000270	2.1980431	0.15082857	2.3654059	254.5632	80.326577	210.76351	54000	3
2000277	2.8871937	0.0873359	1.16195	231.61197	135.57413	264.50517	54000	3
2000287	2.3527778	0.023729805	10.023053	142.48241	120.55732	223.14593	54000	3
2000288	2.7554606	0.2101927	4.32933	120.56933	83.12444	123.7304	54000	3
2000295	2.7959558	0.1703209	2.70624	276.12026	148.56188	226.83366	54000	3
2000296	2.2287841	0.1597919	1.74678	121.59151	252.57079	286.30861	54000	3
2000305	3.1048717	0.1859286	4.44565	207.85299	260.20059	112.25397	54000	3
2000306	2.3578262	0.15049	7.26805	142.03554	167.61862	189.61238	54000	3
2000311	2.8978498	0.0078441	3.22488	81.1549	40.37368	271.97283	54000	3
2000312	2.7808834	0.1616625	9.03534	6.74435	260.2729	0.17345	54000	3
2000321	2.8862686	0.0430939	2.59384	40.46183	30.91652	123.78883	54000	3
2000328	3.1079701	0.1126465	16.07973	352.61251	101.00197	41.63909	54000	3
2000339	3.0128991	0.09462	9.92931	173.79528	160.47059	317.36304	54000	3
2000340	2.7459333	0.1173656	4.67867	27.11612	42.5169	323.31427	54000	3
2000341	2.1992684	0.1937417	5.66779	29.1945	293.41228	8.27634	54000	3
2000346	2.7950112	0.1025065	8.76066	92.16618	289.94435	152.33019	54000	3
2000351	2.7639952	0.156579	9.19393	99.44622	31.66235	228.34098	54000	3
2000352	2.1939318	0.1501057	3.38211	247.42532	144.23938	62.80526	54000	3
2000354	2.8004843	0.11333869	18.379466	140.45317	7.156191	84.076712	54000	3
2000364	2.2209348	0.1490966	6.00474	105.61761	312.93786	210.62896	54000	3
2000374	2.7796261	0.0798103	8.98647	219.23568	27.75383	246.94554	54000	3
2000376	2.2888879	0.1714106	5.43035	302.25603	316.30919	318.79015	54000	3
2000378	2.7767367	0.1295631	7.01027	232.7592	156.09891	167.15252	54000	3
2000384	2.6511976	0.1483651	5.60405	48.10942	33.56521	73.08764	54000	3

2000385	2.8472955	0.12656961	13.565001	345.24165	188.09878	148.57539	54000	3
2000387	2.7391293	0.23705267	18.134397	128.31422	157.68246	180.64512	54000	3
2000389	2.6089236	0.065160176	8.1342143	282.55924	263.55197	227.94975	54000	3
2000394	2.7601348	0.229052	6.22412	67.37124	269.66619	179.60716	54000	3
2000397	2.6347117	0.2465269	12.835543	228.26765	139.38269	93.681871	54000	3
2000402	2.5584357	0.1127408	11.82138	129.5359	18.17371	115.8071	54000	3
2000403	2.8107151	0.0966235	9.15501	244.84291	251.86352	320.86082	54000	3
2000416	2.7913658	0.21853534	12.862244	58.208003	198.84647	225.9835	54000	3
2000421	2.5407221	0.2827883	7.77214	187.53759	209.22374	44.4314	54000	3
2000432	2.3691579	0.1462883	12.13154	88.87357	174.15765	110.98779	54000	3
2000443	2.2155777	0.040032951	4.2304545	175.55548	348.80375	213.66194	54000	3
2000453	2.1829344	0.1089484	5.5578	11.82328	220.16628	299.0741	54000	3
2000458	2.9945128	0.2423835	12.62356	134.95413	274.82091	141.13805	54000	3
2000459	2.6205643	0.2094949	10.29674	29.57945	19.25982	334.1296	54000	3
2000462	2.8740387	0.0830898	3.1911	105.32977	250.36867	293.74567	54000	3
2000470	2.404738	0.0933121	7.22745	173.30542	46.2614	88.91894	54000	3
2000471	2.8861595	0.23353842	14.98526	84.095024	314.48203	42.91019	54000	3
2000472	2.5439924	0.0938108	15.79966	127.26845	296.35407	218.67478	54000	3
2000477	2.4151108	0.1882294	5.28847	10.73814	322.28708	359.09369	54000	3
2000478	3.0152043	0.0882385	13.17546	234.02285	241.42639	277.32205	54000	3
2000480	2.6444378	0.0466305	21.29332	237.39549	212.40356	342.11821	54000	3
2000482	2.9986012	0.1030376	14.46735	179.55439	88.07899	322.25031	54000	3
2000487	2.6712337	0.0864081	10.23447	114.89867	281.04144	229.32471	54000	3
2000496	2.1987751	0.079568	3.78914	207.7666	258.15505	259.49828	54000	3
2000502	2.3826518	0.1791063	25.00994	133.09316	19.42627	26.13608	54000	3
2000509	3.0645243	0.0899784	15.41164	217.81003	157.19742	309.23183	54000	3
2000513	3.0200353	0.0784204	9.71732	184.69574	222.61276	193.18467	54000	3
2000519	2.78994	0.185968	11.01574	44.81205	303.08502	61.36499	54000	3
2000529	3.0167164	0.0952662	11.02327	65.28007	336.21389	338.78737	54000	3
2000532	2.7705832	0.17861371	16.313485	107.60159	76.778716	81.370926	54000	3
2000533	2.9792699	0.0425945	6.55229	180.57574	40.39599	315.61114	54000	3
2000534	2.883731	0.0570623	3.27661	94.2555	333.48245	104.92852	54000	3
2000540	2.2187693	0.0900753	5.57618	202.26254	337.04541	92.34394	54000	3
2000542	2.9068967	0.140981	12.06846	153.25681	214.26388	203.84649	54000	3
2000548	2.2830652	0.1843779	3.87117	108.5126	320.26734	178.898	54000	3
2000549	2.6819981	0.2607464	3.96628	291.64694	156.97231	44.20735	54000	3
2000550	2.5885648	0.2210401	10.11404	270.82874	44.6313	246.80346	54000	3
2000556	2.465813	0.1016516	5.23196	286.23223	177.69124	103.79414	54000	3
2000562	3.0200435	0.0947892	11.12606	70.78883	261.54824	0.27796	54000	3
2000563	2.7115251	0.2362785	10.24831	85.46154	336.61987	31.05904	54000	3
2000565	2.4441963	0.1283881	10.99229	226.06462	290.83268	264.72051	54000	3
2000571	2.4096288	0.2426047	5.22678	3.25658	27.65282	336.72112	54000	3
2000574	2.2520296	0.2397801	5.68496	336.85928	76.87759	280.01942	54000	3
2000579	3.0098865	0.0827351	11.02146	82.83799	231.62261	199.22196	54000	3
2000582	2.6093875	0.2250105	30.01247	155.81846	309.97325	326.65629	54000	3
2000584	2.3743584	0.2329791	10.7257	282.30101	84.784938	276.38474	54000	3
2000599	2.7703969	0.2938211	16.67178	44.68837	292.98145	185.01868	54000	3
2000611	2.9811262	0.1184348	13.44867	189.87939	253.66966	105.58655	54000	3
2000616	2.5543067	0.0578729	14.96176	356.29941	108.68871	107.63951	54000	3
2000619	2.5202746	0.0752591	13.78251	187.56918	178.39417	21.96518	54000	3

2000622	2.416276	0.24156635	8.6414972	142.12698	256.47956	228.26596	54000	3
2000631	2.7905149	0.0854026	18.93284	224.78721	279.07028	175.92013	54000	3
2000633	3.0197599	0.0877737	10.90825	147.54125	185.51247	261.16568	54000	3
2000639	3.0180846	0.1023474	8.57595	280.09687	67.46954	300.6397	54000	3
2000642	3.1960193	0.118776	8.14064	6.78999	112.02326	51.50708	54000	3
2000644	2.6016046	0.1544563	1.04065	109.95042	268.66057	230.99733	54000	3
2000651	3.0236314	0.0963724	10.76903	38.21433	355.83247	296.88522	54000	3
2000653	3.0142155	0.044892	11.28504	133.23306	49.90575	204.14329	54000	3
2000658	2.8545654	0.0617484	1.50589	351.20658	61.9558	222.70176	54000	3
2000660	2.5333023	0.1064984	15.21514	157.15003	104.86147	26.85243	54000	3
2000661	3.0162704	0.0366429	9.25742	336.01605	169.75515	302.94086	54000	3
2000669	3.0126289	0.0824955	10.78161	170.90917	114.17126	319.51092	54000	3
2000673	2.8147227	0.0107688	2.87946	226.90118	256.96914	146.86493	54000	3
2000674	2.9256113	0.1924971	13.51228	58.23889	42.34252	156.94179	54000	3
2000675	2.7677939	0.2042498	9.80224	263.42953	151.95144	49.7925	54000	3
2000686	2.588401	0.269099	15.68124	243.42817	88.35887	104.57222	54000	3
2000695	2.5396531	0.1599762	13.85554	275.79136	79.48882	27.07343	54000	3
2000708	2.6712774	0.0832228	3.48812	355.32832	197.68247	275.39522	54000	3
2000714	2.5353112	0.0572934	14.27153	234.0746	230.23879	331.17093	54000	3
2000716	2.8129656	0.0860072	8.4957	146.29921	53.18106	175.64261	54000	3
2000720	2.8869458	0.0134885	2.3584	35.93197	104.59844	1.17256	54000	3
2000736	2.2019306	0.1649511	4.37435	135.97225	200.35955	320.7405	54000	3
2000737	2.5912677	0.2429092	12.36024	184.98656	133.69582	255.05591	54000	3
2000742	3.0109608	0.119093	11.21532	64.367	286.12007	103.97612	54000	3
2000749	2.2431611	0.1736152	5.38885	109.88041	128.43565	263.27234	54000	3
2000753	2.3290415	0.2213364	10.08948	61.48843	203.04489	68.30372	54000	3
2000770	2.2209539	0.151461	4.3891	44.8096	17.75077	8.33105	54000	3
2000775	3.010628	0.0748552	9.28083	298.00083	169.97933	273.2585	54000	3
2000782	2.1799237	0.0385567	5.26248	80.53587	81.47485	288.82101	54000	3
2000797	2.5346964	0.0602577	4.50134	238.47756	352.71322	72.6863	54000	3
2000800	2.1930117	0.2020272	4.2661	325.26844	347.23211	303.5713	54000	3
2000807	3.0163759	0.0668862	11.30563	132.34846	341.59535	226.7289	54000	3
2000811	2.8951889	0.0757905	3.13615	130.95759	180.32103	194.10173	54000	3
2000824	2.7946416	0.1328303	8.11508	141.74643	140.23895	47.45443	54000	3
2000839	2.6138575	0.1538848	12.60576	338.27524	339.22259	144.00512	54000	3
2000847	2.7838528	0.0939305	2.48025	271.19928	128.06198	166.21445	54000	3
2000851	2.2282453	0.0907124	2.39137	141.24049	7.12275	115.0055	54000	3
2000858	2.8092647	0.1035334	8.88289	67.30295	175.63874	73.08798	54000	3
2000864	2.2082083	0.1899383	5.44451	163.21265	193.85566	326.6475	54000	3
2000876	3.011357	0.1077304	11.3311	151.14979	210.64758	313.11061	54000	3
2000883	2.2379184	0.1993118	4.71591	285.70745	42.16645	232.0517	54000	3
2000888	2.7086863	0.1942403	13.85879	124.23326	297.67398	349.84095	54000	3
2000897	2.5415771	0.0947401	14.32899	258.05353	22.97731	293.86656	54000	3
2000901	2.2237482	0.2215654	3.44429	265.30654	68.06748	201.12121	54000	3
2000925	2.6997365	0.081273206	21.068739	299.73888	201.94338	171.13251	54000	3
2000937	2.231828	0.2178144	3.69552	243.80823	72.00629	311.97017	54000	3
2000939	2.2465816	0.1774328	2.5884	327.28426	5.94831	215.58057	54000	3
2000945	2.6371401	0.162079	32.84988	318.39121	161.12384	7.3489	54000	3
2000951	2.2093123	0.17412152	4.1024509	253.21825	129.49617	113.11692	54000	3
2000962	2.9048251	0.1017686	2.60193	145.66692	225.13159	69.99682	54000	3

2000963	2.2475028	0.1378807	7.98932	62.56217	4.90994	44.92148	54000		3
2000966	2.7208903	0.1277792	14.39219	72.62607	178.16174	144.49032	54000		3
2000968	2.8690829	0.1349555	11.59647	209.00875	298.94054	78.47586	54000		3
2000974	2.5327706	0.111811	5.46304	86.76519	301.95786	106.25134	54000		3
2000975	2.8338758	0.0354774	2.55965	38.85387	52.91719	345.36437	54000		3
2001029	2.8902401	0.022333	2.42891	30.1445	140.88708	327.05951	54000		3
2001043	3.0918753	0.0468296	8.92769	159.56785	157.83728	246.86415	54000		3
2001047	2.2407153	0.1930689	5.66428	78.33893	299.87517	162.4419	54000		3
2001052	2.2358075	0.1440351	4.69483	99.67349	297.39036	80.72568	54000		3
2001055	2.1983055	0.2076132	5.2722	147.21064	176.30191	0.29895	54000		3
2001058	2.1965147	0.187668	3.68964	221.93836	93.99614	318.1367	54000		3
2001078	2.2700948	0.1382134	7.367	93.94718	43.88779	68.71652	54000		3
2001079	2.8768912	0.0437337	1.17674	329.6275	106.23144	163.95851	54000		3
2001087	3.0136918	0.0953399	10.0704	30.51174	28.59489	336.51307	54000		3
2001088	2.2013686	0.1962884	7.65469	54.58213	319.43627	74.07687	54000		3
2001112	3.0214941	0.1014478	8.99516	302.99073	86.60968	259.12752	54000		3
2001129	3.0275955	0.0795809	8.60007	269.61883	134.63107	180.36891	54000		3
2001133	2.1861828	0.1868778	5.37678	58.33863	306.58117	287.54725	54000		3
2001140	2.7718627	0.1109592	14.13293	72.1973	311.2347	288.89636	54000		3
2001148	3.0137128	0.1155158	10.84361	145.68035	174.52707	245.78469	54000		3
2001185	2.2379296	0.1053667	5.70048	71.99598	2.23257	182.64599	54000		3
2001186	3.0185508	0.1082265	10.75844	43.21724	294.1858	222.2892	54000		3
2001215	2.5776575	0.1332289	15.91705	123.81564	265.72624	93.16039	54000		3
2001216	2.2322234	0.1793551	7.60351	121.67187	144.58578	32.44269	54000		3
2001223	2.868483	0.0605204	2.55052	41.07034	10.09496	109.17591	54000		3
2001224	2.3050331	0.1986313	7.87488	258.27234	128.83203	192.44571	54000		3
2001245	2.8931802	0.077307	2.88642	151.89411	207.55894	296.80457	54000		3
2001249	2.2241593	0.0759426	4.87109	259.11704	223.39854	41.71452	54000		3
2001252	2.6956455	0.204297	33.8886	141.08271	62.91168	193.19854	54000		3
2001274	2.2289372	0.1135176	4.39773	327.29125	244.48473	288.76466	54000		3
2001286	3.0223851	0.0884699	9.73977	200.93254	102.70785	4.68589	54000		3
2001289	2.8612633	0.0584571	1.61241	193.25201	117.98548	59.06832	54000		3
2001306	3.1478297	0.096918	14.91028	274.51946	134.41113	160.8342	54000		3
2001307	2.2510661	0.0962503	3.94621	233.97124	207.16335	128.16734	54000		3
2001314	2.2950475	0.1750754	5.24428	264.77378	144.00208	318.51489	54000		3
2001329	2.6180742	0.1709445	14.4656	132.21704	165.03163	26.91872	54000		3
2001336	2.8506578	0.0598015	3.19404	97.51433	218.75154	19.58499	54000		3
2001339	3.019179	0.0560568	8.68943	291.08859	166.96765	224.50772	54000		3
2001350	2.857687	0.0873096	2.93719	139.67262	239.65381	327.23494	54000		3
2001391	2.545694	0.1678673	7.58659	103.57028	85.14437	96.86838	54000		3
2001401	2.2263167	0.1800958	7.28695	277.67193	70.88823	155.25928	54000		3
2001415	2.2234386	0.08697	3.42654	329.4137	240.54671	304.88173	54000		3
2001416	3.0236638	0.102367	10.04353	353.02629	62.48575	166.59297	54000		3
2001418	2.2418907	0.2038894	7.19825	355.20138	324.07708	277.18144	54000		3
2001422	2.2478713	0.1669438	2.67554	201.72553	170.75431	259.86018	54000		3
2001434	3.0188203	0.0609659	10.81365	152.79993	142.70098	60.04093	54000		3
2001442	2.8732162	0.0811964	1.25469	221.18371	126.91285	149.86304	54000		3
2001449	2.2225604	0.1423314	6.63863	110.83068	131.9687	179.36625	54000		3
2001500	2.2420833	0.190578	7.44355	20.04662	16.77771	79.33813	54000		3
2001504	2.3994278	0.1585168	11.04154	94.96816	51.38235	86.57975	54000		3

2001532	3.0037347	0.0553451	8.7858	330.92626	127.1195	284.23106	54000	3
2001533	3.0128199	0.0341753	10.69007	156.87489	359.37335	324.18222	54000	3
2001584	2.3765044	0.1942035	26.6425	305.46984	187.97207	263.98458	54000	3
2001601	2.2338815	0.12969	4.94351	74.77234	196.50973	69.6855	54000	3
2001602	2.2446852	0.1036564	4.16426	75.1916	73.33204	325.00527	54000	3
2001619	2.2410031	0.1755953	6.21376	61.59932	328.19457	270.73248	54000	3
2001621	2.2300159	0.1197028	3.16935	182.00627	238.19504	329.30148	54000	3
2001636	2.234451	0.1280727	4.4334	168.50633	238.94113	63.78235	54000	3
2001644	2.5505794	0.153738	7.01216	270.95555	197.49709	124.28199	54000	3
2001648	2.2356559	0.2070075	4.56623	130.50431	134.30702	151.96224	54000	3
2001657	2.3493066	0.2345214	23.40366	105.44236	53.94026	238.70086	54000	3
2001665	2.4134306	0.2075508	10.8324	91.66581	6.12477	194.44845	54000	3
2001681	2.6953244	0.207426	7.22492	94.62869	0.72186	0.45966	54000	3
2001707	2.2187809	0.1709882	4.03832	6.28529	42.38917	107.0253	54000	3
2001711	3.0143118	0.1116368	11.08177	134.93638	252.71894	330.46795	54000	3
2001717	2.1959125	0.1286555	6.19124	340.64383	115.78748	92.40988	54000	3
2001723	3.0115989	0.0458738	10.92085	150.00584	6.18978	172.39803	54000	3
2001755	3.0902268	0.0498287	10.69528	157.33044	327.98813	229.63128	54000	3
2001830	2.1881582	0.0558756	3.95267	147.57026	335.13972	22.26099	54000	3
2001842	2.2662171	0.1801126	5.35433	153.5978	125.43054	264.22915	54000	3
2001990	2.1741611	0.0512001	3.13114	193.75332	12.25806	241.19664	54000	3
2002000	2.3822032	0.2970119	22.74792	292.18933	129.81927	140.16102	54000	3
2002050	2.3256838	0.2370149	26.57894	72.61229	170.76705	36.08949	54000	3
2002052	3.0071574	0.0836701	9.50768	213.98224	207.86004	301.1236	54000	3
2002089	2.5336949	0.156226	15.39516	102.80752	287.08913	59.54672	54000	3
2002090	3.0715557	0.1351973	11.79893	340.05302	337.43316	329.75184	54000	3
2002111	3.0177442	0.0909077	10.48847	167.35643	233.75672	284.67168	54000	3
2002156	2.2423419	0.2018135	5.35454	17.28794	4.19018	170.84981	54000	3
2002345	3.0156037	0.0784936	9.14384	304.07412	139.06296	287.37178	54000	3
2002411	2.2254446	0.0866982	1.61434	131.05565	129.58459	199.82879	54000	3
2002422	2.3282179	0.1986802	6.40677	160.1336	52.03561	295.16215	54000	3
2002430	2.3627815	0.2139148	23.44678	46.00305	309.62649	16.44569	54000	3
2002510	2.2529475	0.1963001	5.26923	102.98427	209.36488	28.3665	54000	3
2002830	2.3780438	0.2064344	25.32361	49.12056	140.44195	88.0689	54000	3
2012746	2.2382462	0.1908783	4.71175	265.31898	81.63331	100.23076	54000	3

Group	4							
2002062	0.9667013	0.18271178	18.932519	108.62768	147.94205	334.92171	54000	4
2002100	0.83206393	0.4364584	15.757434	170.87687	355.99521	158.8253	54000	4
2002340	0.84389537	0.44995917	5.8539782	211.51958	39.938047	303.69494	54000	4
2003362	0.98946745	0.46856161	9.9183703	152.50975	54.982849	230.97821	54000	4
2003554	0.97371428	0.28048062	23.361458	358.67621	359.38617	297.0219	54000	4
2003753	0.99774049	0.51478951	19.8093	126.29699	43.743147	261.34727	54000	4
2005381	0.94744835	0.29610804	48.973045	58.562459	37.413535	345.61516	54000	4
2005590	0.98568232	0.27958167	14.186004	216.34471	34.416179	218.05496	54000	4
2005604	0.92728217	0.40535123	4.796818	312.00418	82.422924	192.01877	54000	4
2033342	0.71851637	0.41798842	7.3416842	82.004826	167.28181	149.53901	54000	4
2065679	0.91486343	0.26464041	1.2914011	178.35188	14.927526	324.04271	54000	4
2066063	0.99093072	0.72009385	22.666222	351.92132	151.06633	312.1291	54000	4
2066146	0.78727129	0.48380682	5.4099722	102.31935	84.562367	277.80779	54000	4

2066391	0.64231117	0.68842863	38.891521	244.9322	192.59839	260.28645	54000	4
2066400	0.85520974	0.57256097	9.0653163	79.918316	341.27615	303.78632	54000	4
2068347	0.96282557	0.37978055	17.139554	245.97862	243.8006	304.03906	54000	4
2085770	0.99873332	0.3449384	33.178463	18.403672	234.37669	74.743373	54000	4
2085953	0.73876145	0.70312537	12.598508	180.54824	172.32944	193.41695	54000	4
2085989	0.88271351	0.63293609	17.047146	130.29026	309.14389	57.092802	54000	4
2086450	0.9680575	0.41479377	18.106551	124.91312	215.55604	86.689225	54000	4
2086667	0.85926966	0.59473781	14.284786	208.40354	172.40027	208.35792	54000	4
2087309	0.84745188	0.46306937	34.745734	294.31189	188.12379	171.16429	54000	4
2087684	0.85873344	0.64269364	19.234294	162.11324	47.709401	287.41396	54000	4
2088213	0.9539439	0.59522617	17.814259	114.31678	194.94151	172.78666	54000	4
2096590	0.90795755	0.35118792	13.597697	75.804099	202.65036	198.06344	54000	4
2099907	0.72853347	0.59476	28.792699	225.62898	2.8083713	218.4728	54000	4
2099942	0.92226308	0.1910585	3.3313256	204.46	126.39552	84.786507	54000	4
2105140	0.9142539	0.81702607	32.514695	237.49863	281.51518	154.92805	54000	4
3005972	0.89088169	0.66458077	7.2190179	62.010882	193.51103	336.36256	54000	4
3005821	0.90800988	0.27176599	7.2538905	315.473	336.42244	186.28381	54000	4
3092114	0.93557259	0.09336139	12.377796	329.18587	354.30512	193.60748	54000	4
3012397	0.87624175	0.5514582	2.0627017	2.7408046	253.64179	352.82705	54000	4
3092124	0.68441915	0.50211708	3.6341904	197.22215	179.0533	179.7846	49450	4
3005964	0.99299794	0.28400259	23.752266	175.27593	349.68503	252.92589	54000	4
3005969	0.75686007	0.39730327	6.8638137	63.069077	205.68112	168.83864	49709	4
3005970	0.67082068	0.5263641	28.163343	252.69674	356.52092	0.48117591	54000	4
3005973	0.90672544	0.86843062	4.0352333	342.77703	322.39388	99.988219	54000	4
3092144	0.89755766	0.28065043	3.8139909	139.95032	150.27456	206.58537	50107	4
3092156	0.97997322	0.49926135	5.6585162	251.74134	55.816361	72.906521	50427	4
3009717	0.9132051	0.36807412	31.677089	116.97656	141.55437	290.00574	54000	4
3010201	0.93754452	0.34638355	12.773843	260.05979	203.7136	229.81762	54000	4
3010207	0.8655388	0.20828137	16.718068	96.571071	16.625509	314.20271	54000	4
3007848	0.83004916	0.47478051	25.492374	42.445929	180.85609	86.169308	54000	4
3011815	0.89669302	0.3581831	16.208789	344.42506	356.81644	222.69406	54000	4
3013030	0.87849063	0.44056444	3.3998126	53.890616	309.00862	327.02451	54000	4
3013071	0.96311766	0.31262107	7.8072693	183.97257	260.72001	53.391542	54000	4
3092192	0.73134122	0.69855898	30.350464	176.17726	359.82079	118.70808	54000	4
3081066	0.81647942	0.64201038	53.296773	177.26469	359.48387	110.45555	54000	4
3014184	0.70287493	0.5042051	2.9029512	167.15519	6.2466136	188.83079	51077	4
3021790	0.93250655	0.34499429	26.793225	183.98793	35.805081	63.591671	54000	4
3014113	0.81932815	0.53002281	21.049912	197.61161	322.42211	146.69697	54000	4
3014114	0.9032239	0.50379849	23.425461	166.83655	47.499929	92.326803	51081	4
3017039	0.87579594	0.3181469	21.802114	46.44937	170.67926	55.367825	54000	4
3015691	0.85113036	0.44357995	23.750628	236.336	320.85172	257.14348	54000	4
3016523	0.74123992	0.36744228	6.9691489	74.57572	152.86115	301.0636	54000	4
3027730	0.87828555	0.73912509	13.432252	280.10884	353.0528	273.61263	54000	4
3064315	0.98185794	0.2096551	7.2453491	85.995719	226.33812	105.51781	54000	4
3017309	0.91140605	0.11077323	2.6226597	313.34135	7.6536829	82.979128	54000	4
3017060	0.93722796	0.23454007	6.5601162	327.40619	299.48051	33.408243	54000	4
3020946	0.81902244	0.46252854	25.657332	155.93408	253.33986	114.23065	54000	4
3092226	0.90718857	0.33269924	11.907748	240.07869	223.55372	286.28021	54000	4
3019650	0.67402486	0.66520869	2.0164845	80.792574	9.8818082	236.01441	54000	4
3092245	0.92884396	0.1120346	10.776168	232.31222	354.5409	62.090196	54000	4

3024030	0.89999766	0.13956765	1.6633228	55.303545	151.72637	221.34283	54000	4
3029428	0.82941678	0.55831156	16.741231	349.66292	292.73797	117.89073	54000	4
3025763	0.85357291	0.28634449	4.6958454	101.79881	187.91717	3.8151398	54000	4
3025764	0.87831922	0.41135064	2.6926995	110.86572	200.06223	197.96191	54000	4
3025765	0.74678009	0.36001309	8.6005618	277.58834	7.9287065	310.47211	54000	4
3031020	0.876503	0.89503754	25.676583	333.80753	324.24248	205.0929	54000	4
3031703	0.74122654	0.35849219	6.8900577	70.682966	247.33611	283.11028	54000	4
3028808	0.86324958	0.42308796	3.2721008	214.29152	109.00818	253.08418	54000	4
3031176	0.89556289	0.49535956	11.560793	162.90292	139.56794	175.79773	54000	4
3031177	0.83508053	0.56664333	13.775537	3.9767428	310.03182	116.79868	54000	4
3031178	0.66184819	0.53291455	26.470152	155.80371	197.79925	234.65129	54000	4
3092253	0.81607747	0.46980368	3.8734528	345.26231	23.964009	77.21737	54000	4
3031183	0.94696812	0.12351254	22.321499	331.20642	46.371289	169.54532	54000	4
3031186	0.93766812	0.32108762	5.4192095	178.31731	125.30451	330.94723	54000	4
3035165	0.92871575	0.44688014	40.260068	358.52887	313.90075	275.64076	51624	4
3035166	0.92969018	0.45572011	28.578789	52.953803	277.98821	281.16831	54000	4
3039898	0.75784422	0.47657125	32.143436	358.1575	16.943112	78.616265	54000	4
3036363	0.81552756	0.43044014	2.6620264	55.249024	17.867756	355.02601	54000	4
3092260	0.74399955	0.52412868	5.9877829	30.318292	6.0866893	173.17855	51662	4
3042555	0.91610235	0.11212565	2.8299469	72.769203	7.7459151	237.46661	54000	4
3046648	0.98475099	0.22113264	9.9847342	304.64098	166.10892	312.02601	54000	4
3053717	0.87266158	0.37357638	51.180869	124.4379	7.6049849	322.8322	54000	4
3092272	0.8258742	0.55130757	19.643821	177.92578	354.37058	189.62089	54000	4
3061547	0.95118128	0.31832272	16.093766	312.84998	211.71057	12.786194	54000	4
3050515	0.81138241	0.46688594	10.355937	350.69072	224.30388	170.08819	54000	4
3054338	0.92940488	0.16742376	0.89619274	14.741749	131.29453	116.32414	54000	4
3062815	0.87028331	0.4223829	32.228534	29.823765	187.37091	271.42008	54000	4
3054373	0.88325596	0.24874524	0.7818067	237.8882	293.04609	279.96198	54000	4
3063789	0.90366249	0.43874889	11.744223	33.888091	228.71238	38.109895	54000	4
3063058	0.87949364	0.26271471	17.412879	50.862594	229.98231	141.9183	54000	4
3063823	0.85448153	0.28864592	7.6757121	55.931647	221.95088	252.14788	54000	4
3057545	0.91136333	0.78064631	7.783026	69.374003	213.58275	302.66343	54000	4
3067492	0.85678956	0.22462344	3.4909518	97.371774	189.32344	247.32303	54000	4
3067616	0.8233799	0.3689377	17.513089	297.87042	30.585412	332.21202	54000	4
3068066	0.9402947	0.13738105	5.7684736	115.62224	242.83832	185.9995	54000	4
3071939	0.8542365	0.17238766	2.0261914	122.57094	195.57522	217.07665	54000	4
3072413	0.72538055	0.38255603	8.1364752	109.55428	234.13343	173.96749	54000	4
3069758	0.71449449	0.4073083	10.551364	331.00521	353.49934	284.26514	54000	4
3070801	0.93967934	0.17625793	1.2921429	31.947763	342.489	166.51755	54000	4
3072196	0.98914235	0.057313022	11.62947	357.82492	306.54524	94.254912	54000	4
3076722	0.94421086	0.60426198	20.663998	22.148342	339.94382	205.64386	54000	4
3072273	0.98269502	0.02797089	5.2439975	183.09711	233.5423	352.59284	54000	4
3072291	0.88860879	0.15914197	7.2897211	189.24157	200.81708	356.38095	54000	4
3076775	0.87462622	0.49934979	23.74621	32.651464	28.155709	337.96255	54000	4
3074756	0.91403391	0.41209793	5.209049	205.39319	210.98323	203.67515	54000	4
3079950	0.93386971	0.32328704	12.090374	295.94754	142.88185	155.09714	54000	4
3089251	0.89155077	0.21371778	50.207445	317.70623	244.31983	122.67802	54000	4
3092324	0.95860272	0.240769	13.877065	170.77281	330.28055	356.41634	54000	4
3092325	0.91409916	0.34251754	7.5207799	154.15697	4.3166007	117.71373	54000	4
3092357	0.90650678	0.39299432	17.604042	11.3038	213.96837	38.677025	54000	4

3092339	0.94807271	0.49150795	3.951064	327.3059	262.35483	225.1088	54000		4
3092347	0.95414371	0.16607503	9.011133	13.218041	241.34418	217.30782	54000		4
3092370	0.91129913	0.5259771	31.336012	27.399044	208.76826	338.40154	54000		4
3092377	0.9619541	0.48147436	19.038812	12.90751	199.00974	201.23957	54000		4
3092380	0.87475176	0.54598912	15.203483	57.856362	135.96125	231.29666	54000		4
3092386	0.79674105	0.77741935	25.419665	30.332708	212.38759	52.895742	54000		4
3092390	0.88493377	0.28666987	7.7052214	25.592661	133.00062	225.31406	54000		4
3102665	0.75103189	0.3734296	18.179766	239.71613	358.3795	188.25934	52242		4
3102680	0.79739141	0.54626241	27.159807	69.743317	208.48745	19.485949	54000		4
3102687	0.87176578	0.38714096	30.995333	92.982658	219.67403	59.371016	54000		4
3102718	0.67636878	0.54206925	4.8003637	306.08255	318.32119	221.27778	52271		4
3102727	0.8799063	0.54157069	33.006701	294.73391	25.324129	351.02314	54000		4
3102728	0.77873106	0.43775061	29.88594	287.90059	323.84803	192.59727	54000		4
3102731	0.84083923	0.38595466	13.181771	103.14846	149.5432	282.4749	54000		4
3102744	0.85560809	0.3736	17.179764	99.516019	205.13541	153.88991	54000		4
3102756	0.91555629	0.16340646	13.072181	295.18464	306.60939	16.12534	54000		4
3102762	0.99426257	0.013065778	10.742631	106.46935	100.60964	164.92244	54000		4
3102779	0.87507581	0.54652018	27.746329	115.8071	147.21096	156.49332	54000		4
3102787	0.97987696	0.1766903	6.879208	8.7510894	331.56631	51.187861	54000		4
3114017	0.97888269	0.42840825	2.4597657	81.442256	272.77239	357.32802	54000		4
3114023	0.86557884	0.22561468	3.1333566	137.63141	210.35146	42.307944	54000		4
3114026	0.8198196	0.40303203	12.607732	137.70231	217.99794	186.56678	54000		4
3114104	0.8577063	0.36944345	16.599775	234.34539	94.099107	12.146551	54000		4
3117427	0.92124305	0.36305932	1.5475656	347.22658	57.667418	131.23247	54000		4
3120884	0.92160627	0.56640928	30.14657	262.94414	25.307036	177.07735	54000		4
3117446	0.82348932	0.341939	6.5978745	164.19778	223.15648	134.38613	54000		4
3117447	0.76147828	0.60181997	20.270204	203.66158	148.24457	40.981494	54000		4
3117460	0.96703756	0.30044155	28.064829	7.9945393	31.452424	26.207003	54000		4
3117468	0.98825788	0.46273267	9.4897329	188.61692	226.69016	281.87596	54000		4
3120861	0.99226321	0.52902156	22.554197	40.865148	8.3075997	259.155	54000		4
3120863	0.76830954	0.37557174	10.628605	189.48132	206.28661	4.8467488	54000		4
3125004	0.81884734	0.39093655	40.85275	69.427745	306.91181	146.81126	54000		4
3124996	0.7701479	0.30533751	4.3155863	68.645093	338.18528	337.37454	54000		4
3125009	0.89868318	0.26626348	11.765305	218.53494	175.17419	259.49608	54000		4
3126183	0.92470305	0.29777782	3.7633964	203.56944	253.41942	71.988181	54000		4
3127391	0.95500888	0.37938048	2.9089622	248.23483	133.95352	127.29616	54000		4
3127401	0.71992412	0.49553466	0.76027896	166.75667	282.02629	166.8299	54000		4
3127406	0.84475671	0.31396358	6.1999594	259.60686	162.73931	25.047051	54000		4
3130459	0.9135579	0.27422333	36.28443	109.10504	346.86651	103.49846	54000		4
3131055	0.87651922	0.43432919	5.4175723	259.65972	222.15883	198.16622	54000		4
3132092	0.93591413	0.24289976	6.9056071	174.41506	318.27888	347.97534	54000		4
3133156	0.81697506	0.69905755	12.748378	164.33324	355.26495	358.67083	54000		4
3134264	0.96703422	0.30996246	13.538525	349.95955	156.04858	305.86572	54000		4
3134268	0.82550846	0.28643033	1.3250455	92.022439	71.690414	355.44719	54000		4
3136734	0.90470877	0.60068666	20.869143	350.95397	169.36365	127.32394	54000		4
3137844	0.93000656	0.12057551	8.4784677	13.140614	223.07557	275.96035	54000		4
3141527	0.79876331	0.48723132	30.700644	209.35617	358.62917	331.99547	54000		4
3141535	0.83739563	0.43662198	9.6948341	222.32902	348.73063	247.83159	54000		4
3141538	0.7236569	0.41051242	8.9801315	231.66157	355.51291	280.154	54000		4
3143084	0.98456037	0.20136283	2.331352	216.87263	78.254887	72.980025	54000		4

3143121	0.90584107	0.23739246	25.53355	245.93098	351.2243	346.24144	54000	4
3144153	0.95516322	0.35943369	21.528325	265.95557	317.58094	263.35691	54000	4
3144155	0.90911752	0.21719253	2.0854213	160.69647	118.5063	274.21168	54000	4
3144531	0.80953528	0.24208909	34.160061	81.700317	178.61437	246.44399	54000	4
3145517	0.87611211	0.38410512	7.3889096	301.85306	23.557751	298.6393	54000	4
3146499	0.87482965	0.42612924	23.236807	286.83005	43.944275	283.29971	54000	4
3147579	0.9203212	0.11970839	7.4793473	139.95971	172.95921	223.42272	54000	4
3150768	0.95752824	0.051791757	15.269684	346.05269	23.978949	75.424192	54000	4
3150774	0.93470009	0.24963774	13.220376	178.04024	167.88093	155.74913	54000	4
3151641	0.70741718	0.48594178	23.375604	177.59264	196.51278	158.54796	52724	4
3151644	0.8585099	0.39390777	13.050528	21.684072	339.2499	342.06502	54000	4
3151655	0.73081812	0.58168944	6.6462784	359.37936	29.591842	322.20328	54000	4
3152309	0.89306537	0.21856995	12.031238	196.34869	181.81855	202.52777	54000	4
3152317	0.87232869	0.18199623	17.024551	199.60692	168.68648	275.52077	54000	4
3153508	0.84990274	0.380551	18.108305	70.456253	306.68814	322.20206	54000	4
3153509	0.81317508	0.27040152	26.275789	30.512696	19.241924	24.021864	54000	4
3153530	0.81508903	0.26213285	4.8810454	39.08647	351.8594	56.376705	54000	4
3154503	0.72744426	0.5109053	23.505997	215.53035	203.9569	308.37372	54000	4
3154513	0.94882172	0.33060063	23.901897	250.048	154.67071	73.518182	54000	4
3154520	0.96054013	0.14975606	10.795804	247.33008	238.12675	294.20537	54000	4
3156302	0.85723098	0.21046138	0.63241404	215.81727	210.40318	248.98378	54000	4
3157339	0.79330244	0.49250313	18.233456	124.69036	311.55026	87.736143	54000	4
3160723	0.88859382	0.18325773	4.6515135	178.77306	316.20899	88.522675	54000	4
3160748	0.88518207	0.30354972	3.6597329	176.57871	47.968282	321.24813	54000	4
3160799	0.82842354	0.2097393	8.4615437	274.20954	326.31267	130.70576	54000	4
3160853	0.90779594	0.31592396	25.432456	200.72375	355.12527	314.04643	54000	4
3163736	0.77653201	0.38182999	12.146048	220.16201	321.8564	329.76908	54000	4
3164401	0.70092845	0.59594346	16.508885	22.958041	200.59164	116.86678	54000	4
3164404	0.78130319	0.3368181	3.794778	188.87281	59.273754	206.13498	54000	4
3164431	0.97915383	0.14704531	17.058952	212.81145	287.43237	269.39447	52939	4
3167348	0.90863991	0.5445115	28.539008	57.597699	140.64874	280.45945	52966	4
3167353	0.99058994	0.12109868	2.5230027	42.306512	225.08689	97.336963	54000	4
3167367	0.88991985	0.18053258	0.3560634	65.466287	138.0166	5.1901772	54000	4
3170202	0.93049583	0.19888123	19.460508	89.811876	165.60388	236.09578	54000	4
3170242	0.89864059	0.45058732	29.268846	86.451474	138.05774	357.61805	54000	4
3170203	0.87876494	0.26655962	5.7547748	89.91715	222.81766	232.34244	54000	4
3170204	0.93033425	0.31303949	6.5240763	99.18043	134.48003	274.13873	54000	4
3170221	0.99414961	0.013466245	4.3015637	265.1633	83.895555	14.977607	54000	4
3170208	0.96902587	0.35503538	2.735198	86.544614	127.98154	235.18049	54000	4
3172322	0.88403687	0.22188425	3.6171905	299.09124	28.23779	217.61998	54000	4
3174187	0.96090346	0.38494142	17.667309	300.91031	45.800414	38.593947	54000	4
3175337	0.94402813	0.40024487	23.023004	157.35712	216.07589	41.130168	54000	4
3176187	0.88455581	0.33025347	5.141462	336.708	50.031926	128.14063	54000	4
3177176	0.98950948	0.27977013	4.6639542	343.45054	55.822201	302.36837	54000	4
3177188	0.88044837	0.50520426	28.587726	161.13348	202.3902	144.08901	54000	4
3177193	0.81459703	0.26847244	7.5899015	356.20573	337.56064	8.6811993	54000	4
3177197	0.90031102	0.17108277	7.9562364	357.43335	343.24023	181.87279	54000	4
3177202	0.81796821	0.28900036	0.021373399	296.18068	31.319869	14.154055	54000	4
3177226	0.88559948	0.24938109	6.7662088	170.07059	196.2384	197.01913	54000	4
3177232	0.87856871	0.49234168	3.5104099	183.39904	142.00897	280.59227	54000	4

3177234	0.91350452	0.34920303	33.467907	195.47359	210.26718	55.132809	54000	4
3249978	0.82681962	0.39218262	4.1644361	191.24861	139.79199	262.65627	53100	4
3179349	0.69653246	0.48849431	14.547286	115.75337	278.56009	230.04016	54000	4
3179363	0.78916702	0.59877773	28.975304	203.02793	159.32865	47.977419	54000	4
3180192	0.97997642	0.22337695	11.135007	214.71419	112.10254	87.29048	54000	4
3182186	0.95274176	0.56156984	14.734804	235.26374	207.42161	290.45178	54000	4
3182187	0.90121009	0.26577903	10.52491	101.99185	348.94624	27.260244	54000	4
3182823	0.83006288	0.40744127	1.9081961	243.48958	213.17629	141.06958	54000	4
3182829	0.96114304	0.17028425	35.070972	78.132599	350.0398	355.76643	54000	4
3182833	0.71192116	0.49856275	22.055369	79.22656	340.67122	148.13048	54000	4
3183847	0.91488208	0.3513503	25.453919	82.670132	309.52078	121.70682	54000	4
3184475	0.95062062	0.56280186	29.334285	263.9296	231.36764	230.38682	54000	4
3249980	0.95021121	0.4167979	14.339061	327.20744	232.1766	88.266623	53241	4
3250193	0.95199695	0.17441825	56.22508	151.86061	9.1122288	165.27237	53244	4
3250195	0.94312385	0.33803687	6.2477966	75.46293	109.19721	210.80311	54000	4
3250293	0.95092007	0.12170564	0.57411596	175.15212	28.533305	202.67682	54000	4
3251510	0.90403261	0.65680238	15.91456	119.11494	65.154363	317.11312	54000	4
3251512	0.92044085	0.35110701	5.9585478	173.8943	333.85506	323.2308	54000	4
3252104	0.96120257	0.32880847	5.3345585	199.45033	280.81436	315.18661	54000	4
3253645	0.95427638	0.1863123	22.053165	356.79811	226.8361	173.13211	54000	4
3254500	0.87511021	0.46481928	21.334924	46.654234	94.378224	53.416337	54000	4
3255174	0.73778019	0.41619723	18.418049	180.36366	359.49238	183.96342	53272	4
3255464	0.86591757	0.23763089	18.701611	302.13149	233.48043	3.3642175	54000	4
3255465	0.7670961	0.42892764	4.7613942	202.37822	322.48787	207.26302	54000	4
3255879	0.90794321	0.24989018	13.512538	14.576143	200.48313	239.92139	54000	4
3256321	0.75098496	0.44284208	2.6217937	48.487884	136.67192	208.67795	54000	4
3256580	0.89500976	0.2095971	19.353928	155.44787	77.702026	230.72107	54000	4
3256583	0.97640702	0.16192739	36.517136	12.606993	138.64939	238.04986	53289	4
3257077	0.94624489	0.25852582	14.059494	203.10884	56.491094	130.79876	54000	4
3258062	0.95420397	0.39678012	3.7129817	29.862271	120.62117	303.80802	54000	4
3258076	0.9644237	0.22112866	4.5080307	211.98291	294.21659	275.85009	54000	4
3261401	0.94051139	0.2439765	16.218207	225.83194	297.07694	286.7039	54000	4
3261681	0.94377277	0.16445192	1.2936976	233.53862	332.35792	220.29651	54000	4
3262331	0.96831267	0.6554068	36.276971	208.92346	43.876742	42.361252	54000	4
3262569	0.85560093	0.16979483	10.348597	54.94469	179.78693	299.55207	54000	4
3263232	0.83752123	0.2980902	1.2031678	285.35506	0.81535776	244.86104	54000	4
3263233	0.88711985	0.1708655	12.187234	253.39186	15.930031	209.35138	54000	4
3263448	0.74868835	0.43382753	3.0995667	307.15697	302.8351	107.2211	54000	4
3263449	0.76001344	0.40978133	21.488824	85.680513	157.51318	99.054976	54000	4
3263451	0.93149302	0.26650978	10.738706	120.95563	115.66393	228.79036	54000	4
3263793	0.64024467	0.7968061	23.746543	122.70088	130.7697	23.609494	54000	4
3264188	0.86840745	0.31327764	6.0667175	263.47717	47.285452	167.70475	54000	4
3264189	0.84262456	0.24011097	12.086974	265.61112	7.8051078	264.56936	54000	4
3264547	0.80883312	0.5361478	28.149668	269.94609	31.209738	243.68826	54000	4
3265905	0.872222	0.56873666	5.8691155	117.66371	155.76728	265.87081	54000	4
3265909	0.88383727	0.42113866	31.187501	116.00428	168.67007	191.20433	54000	4
3266031	0.84659761	0.30073285	12.969629	296.89813	38.72732	170.98457	54000	4
3266035	0.94878176	0.35599452	10.674391	113.41088	174.10611	131.31082	54000	4
3267564	0.99121273	0.068706638	9.5265808	147.02475	248.61695	326.2324	54000	4
3273458	0.89266375	0.77031035	16.320488	344.39249	328.04398	297.39556	53433	4

3273788	0.95945314	0.13537803	30.002363	329.84879	347.06403	8.5373459	54000	4
3273782	0.76389989	0.38338784	20.484115	353.00599	351.28138	192.40885	53441	4
3274691	0.91854667	0.27307317	12.946712	0.23650582	310.74842	150.4935	54000	4
3274905	0.93308734	0.33019919	3.7480425	177.42193	120.84034	160.1944	54000	4
3275869	0.7532051	0.34020433	24.917136	272.74031	156.55353	293.86874	54000	4
3275978	0.77884535	0.38380306	28.00374	22.727001	334.54877	268.0763	54000	4
3276398	0.95886166	0.24585539	5.5684355	229.95645	112.70238	75.266559	54000	4
3276601	0.79131389	0.39482129	9.1506832	161.0855	243.61206	140.13948	54000	4
3276686	0.95138211	0.13570386	18.653347	203.11495	230.49457	316.55945	54000	4
3277400	0.85478599	0.33557573	7.8992289	59.628165	6.0811678	42.263504	54000	4
3278402	0.84051334	0.21493604	2.9048383	226.50305	181.14322	102.53286	54000	4
3279867	0.98527456	0.79257712	41.420794	88.69818	42.781343	197.64349	54000	4
3283218	0.80376703	0.38177862	29.514513	98.090608	349.71695	85.132792	54000	4
3283227	0.85281863	0.29558723	27.788514	263.65459	190.52382	9.5232994	54000	4
3283249	0.86343014	0.41102769	6.3146007	176.759	250.09525	64.949008	54000	4
3283679	0.7892811	0.49636097	10.639743	289.8331	194.62252	51.811201	54000	4
3283835	0.77941341	0.48307519	6.0356351	114.63866	0.58825253	89.042008	54000	4
3283950	0.86926394	0.42248981	26.58707	120.89719	1.6808852	337.42373	54000	4
3285073	0.97601434	0.31997874	12.52351	309.81948	204.75039	202.77323	54000	4
3288855	0.89356017	0.36457579	9.4605918	48.218676	108.6557	278.29264	54000	4
3288933	0.97558635	0.17580176	3.9570556	334.9335	119.6269	297.85286	54000	4
3289173	0.99925501	0.30304443	33.871509	155.09527	54.330821	115.24323	54000	4
3289739	0.87749296	0.39370241	36.068592	165.88104	320.37879	349.70934	54000	4
3290865	0.98106367	0.28308734	34.840215	23.566708	205.51961	110.17821	54000	4
3291224	0.8655333	0.62371237	27.756769	354.90012	228.05435	158.12306	54000	4
3292261	0.84131167	0.41651256	5.2041768	8.67502	151.88261	240.44876	53647	4
3293790	0.82682197	0.23366556	25.622782	14.945903	185.79892	267.99709	54000	4
3293831	0.94924874	0.37694757	5.0055771	195.25531	304.934	276.71907	53652	4
3293922	0.92386804	0.13387525	2.4267858	346.0492	199.58911	220.00037	54000	4
3293923	0.83735026	0.22569769	0.73313176	196.83563	18.129628	155.00452	53655	4
3297182	0.89288955	0.17018689	5.6532049	32.652164	139.63883	260.57564	54000	4
3297356	0.93729026	0.56966238	14.293632	58.989362	127.58469	263.15198	54000	4
3297379	0.95814923	0.11600732	5.4166472	216.10028	313.89196	220.99061	54000	4
3297628	0.74145111	0.42653472	24.545328	223.74752	358.53892	309.99123	54000	4
3297629	0.89095046	0.22521846	0.2501009	37.454728	228.7358	132.91306	54000	4
3299721	0.94479231	0.23311402	2.0866116	49.351328	115.03663	247.96876	54000	4
3304566	0.6716719	0.57517628	23.030433	69.435747	176.04117	9.3177124	54000	4
3305028	0.95853738	0.15192594	21.623399	288.1221	297.85814	119.57933	54000	4
3306579	0.96807507	0.32044152	32.651067	75.70229	238.56512	87.195098	53713	4
3307228	0.84064591	0.26639019	17.248864	84.890649	149.93067	220.28289	54000	4
3307229	0.78419138	0.41420507	16.853707	282.3189	9.2001035	165.71215	54000	4
3309039	0.7109288	0.55050252	19.581348	288.69066	327.84503	281.67673	54000	4
3309091	0.76032651	0.371895	12.795812	274.41391	20.046772	181.31049	54000	4
3309092	0.81856961	0.27264748	3.6070987	71.203243	222.54974	148.62055	54000	4
3309828	0.74387078	0.33313323	22.196585	282.7454	339.97538	249.41827	54000	4
3309832	0.8212713	0.2963478	15.586801	281.79624	328.32981	221.01954	54000	4
3309857	0.77172747	0.32160915	7.7314462	100.74026	189.88574	184.97114	54000	4
3309858	0.9229554	0.51225824	14.137932	127.80299	191.7352	14.612568	54000	4
3311964	0.98261634	0.64901708	4.1646477	123.37688	139.52651	173.2487	54000	4
3313739	0.91259519	0.36592669	8.3160351	305.98482	25.497647	43.203772	54000	4

3314789	0.81990323	0.4222185	24.380862	146.84225	153.17094	103.18551	54000	4
3315649	0.67622812	0.75501232	10.296226	303.38991	29.482862	217.22446	54000	4
3324253	0.86368636	0.33670256	26.531002	162.3011	187.48359	57.179409	54000	4
3324656	0.69549487	0.49762915	1.7812336	336.43224	17.429135	140.24267	53794	4
3328632	0.92250426	0.34340855	14.622177	15.228858	3.1455032	356.30048	54000	4
3329255	0.95446956	0.19860002	1.5906721	280.91861	154.81012	276.65717	54000	4
3329278	0.95908392	0.17937767	10.060842	183.9189	242.88627	281.19884	54000	4
3330155	0.84194719	0.31558662	31.806967	35.992828	317.68035	246.80424	53846	4
3330538	0.98525825	0.26341245	9.5375754	232.82844	212.56695	246.14269	54000	4
3330688	0.84884937	0.26080106	16.418388	34.794089	22.616138	331.26257	54000	4
3333079	0.67190303	0.58163186	5.9724883	41.039218	17.670959	14.966162	54000	4
3337162	0.83860414	0.60517513	27.26727	291.85793	174.50168	318.38756	54000	4
3337325	0.84773943	0.57581932	20.07954	115.27511	29.325849	199.74045	54000	4
3338368	0.80375651	0.28451361	3.439796	4.9652624	124.76779	235.02335	54000	4
3339082	0.98674925	0.046543451	2.8266463	163.3305	332.95836	225.00633	54000	4
3341199	0.95075741	0.30076367	1.4144836	93.52479	110.27498	136.8701	54000	4
3342642	0.90638531	0.23131827	23.88335	270.9998	261.22068	144.12389	54000	4
3342322	0.80476877	0.34642503	4.8343905	347.5677	182.91696	190.23549	54000	4
3342323	0.94928888	0.28045436	5.865093	228.17418	305.53284	190.56084	54000	4
3343104	0.88152606	0.29156747	4.5585453	358.58023	165.97431	205.79041	54000	4
3344169	0.92191222	0.32902583	32.483792	1.2876999	224.38265	103.38686	54000	4
3347493	0.98577785	0.17475527	2.6419584	194.39292	38.266669	110.3162	54000	4
3348144	0.94018863	0.39616796	11.569969	195.46581	315.13044	265.13542	54013	4
3350632	0.94745779	0.58305705	5.5041004	225.42431	299.84819	272.59296	54021	4
3350633	0.85107872	0.46865622	2.9133876	92.154235	68.436379	235.83518	54000	4

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B: M-FILES USED FOR FEASIBLE SOLUTION

Cost Function:

```
function [eventCost, runningCost] = SPCost(primal)

% Cost File.
% By Dr. Pooya Sekhavat
% Last updated May 31 2007 by David Koeppel

t_0 = primal.nodes(1);
t_f = primal.nodes(end);

m_0 = primal.states(4,1);
m_f = primal.states(4,end);

eventCost   =  (t_f - t_0)/m_f; %m_0 - m_f; % %-(m_f - m_0)/(t_f -
t_0);
runningCost = 0;
```

Dynamics Function

```
function residuals = OrbitDynamics(primal)

% Dynamics File
% By Dr. Pooya Sekhavat
% Last updated May 31 2007 by David Koeppel

global CONST
global EARTH
global ast_ephem
global SCALED

% Spacecraft
x_sp = primal.states(1,:);          x_spdot = primal.statedots(1,:);
y_sp = primal.states(2,:);          y_spdot = primal.statedots(2,:);
z_sp = primal.states(3,:);          z_spdot = primal.statedots(3,:);
m_sp = primal.states(4,:);          m_spdot = primal.statedots(4,:);

vx_sp = primal.states(5,:);         vx_spdot = primal.statedots(5,:);
vy_sp = primal.states(6,:);         vy_spdot = primal.statedots(6,:);
vz_sp = primal.states(7,:);         vz_spdot = primal.statedots(7,:);

thrust_x = primal.controls(1,:);      % Thrust Tx
thrust_y = primal.controls(2,:);      % Thrust Ty
thrust_z = primal.controls(3,:);      % Thrust Tz
thrust_mag = primal.controls(4,:);    % Thrust magnitude
r_mag = primal.controls(5,:);         % Distance from sun
```

```
residuals = primal.states;

%==========================================================
% Equations of Motion:
residuals(1,:) = x_spdot - ((SCALED.Vx*SCALED.Ti)/SCALED.X).*vx_sp;
residuals(2,:) = y_spdot - ((SCALED.Vy*SCALED.Ti)/SCALED.Y).*vy_sp;
residuals(3,:) = z_spdot - ((SCALED.Vz*SCALED.Ti)/SCALED.Z).*vz_sp;
residuals(4,:) = m_spdot  +
(SCALED.TT*SCALED.Ti/SCALED.MU).*thrust_mag./(CONST.g*CONST.isp);
residuals(5,:) = vx_spdot +
((SCALED.X*SCALED.Ti*CONST.mu)/(SCALED.Vx*SCALED.RU^3)).*x_sp./(r_mag.^
3) - ((SCALED.Tx*SCALED.Ti)/(SCALED.Vx*SCALED.MU)).*thrust_x./m_sp;
residuals(6,:) = vy_spdot +
((SCALED.Y*SCALED.Ti*CONST.mu)/(SCALED.Vy*SCALED.RU^3)).*y_sp./(r_mag.^
3) - ((SCALED.Ty*SCALED.Ti)/(SCALED.Vy*SCALED.MU)).*thrust_y./m_sp;
residuals(7,:) = vz_spdot +
((SCALED.Z*SCALED.Ti*CONST.mu)/(SCALED.Vz*SCALED.RU^3)).*z_sp./(r_mag.^
3) - ((SCALED.Tz*SCALED.Ti)/(SCALED.Vz*SCALED.MU)).*thrust_z./m_sp;
```

Events Function

```
function eventConditions = SPEvents(primal)

% Events File.
% By Dr. Pooya Sekhavat
% Last updated May 31 2007 by David Koeppel

global CONST
global EARTH
global ast_ephem
global data
global SCALED
global AstrNo

if (numel(ast_ephem)<1)
ast_ephem = dlmread('ast_ephem.txt');
end

% Spacecraft States x = [x, y, z, vx, vy, vz]
x0     = primal.states(1,1);                xf     =
primal.states(1,end);
y0     = primal.states(2,1);                yf     =
primal.states(2,end);
z0     = primal.states(3,1);                zf     =
primal.states(3,end);
m0     = primal.states(4,1);
vx0    = primal.states(5,1);               vxf    =
primal.states(5,end);
vy0    = primal.states(6,1);               vyf    =
primal.states(6,end);
vz0    = primal.states(7,1);               vzf    =
primal.states(7,end);
```

```
t_0 = primal.nodes(1);
t_f = primal.nodes(end);

m_0 = primal.states(4,1);
m_f = primal.states(4,end);

E0 = primal.parameters (1);     %Eccentric Anomaly
Ef = primal.parameters (2);

t0 = primal.nodes(1)* SCALED.Ti+CONST.MinLaunchJD;
tf = primal.nodes(end)* SCALED.Ti+CONST.MinLaunchJD;


M_Earth = EARTH.M0+sqrt(CONST.mu/EARTH.a^3)*(t0-EARTH.t0);  %Mean
anomaly of Earth

theta = 2 * atan (tan(E0/2) / sqrt((1 - EARTH.e)/(1 + EARTH.e)));
gamma = atan ((EARTH.e * sin (theta))/(1 + EARTH.e*cos(theta)));
%Flight path angle
r = EARTH.a * (1 - EARTH.e^2) / ( 1 + EARTH.e*cos(theta));
%Motion of Earth
v = sqrt(CONST.mu * ((2/r) - (1/EARTH.a)));
%Motion of Earth

%Motion of Earth converted into Cartesian Coordinates
x = r*(cos(theta + EARTH.w)*cos(EARTH.lan) - cos(EARTH.inc)*sin(theta +
EARTH.w)*sin(EARTH.lan));
y = r*(cos(theta + EARTH.w)*sin(EARTH.lan) + cos(EARTH.inc)*sin(theta +
EARTH.w)*cos(EARTH.lan));
z = r*(sin(theta + EARTH.w)*sin(EARTH.inc));
vx = v*(-sin(theta + EARTH.w - gamma)*cos(EARTH.lan) -
cos(EARTH.inc)*cos(theta + EARTH.w - gamma)*sin(EARTH.lan));
vy = v*(-sin(theta + EARTH.w - gamma)*sin(EARTH.lan) +
cos(EARTH.inc)*cos(theta + EARTH.w - gamma)*cos(EARTH.lan));
vz = v*(cos(theta + EARTH.w - gamma)*sin(EARTH.inc));

X_init_Earth =
[x/SCALED.X,y/SCALED.Y,z/SCALED.Z,vx/SCALED.Vx,vy/SCALED.Vy,vz/SCALED.V
z,v];  %Initial position and velocity of Earth

eventConditions = zeros(7, 1);

eventConditions(1) = 1e5*(((x0 -
X_init_Earth(1))*(SCALED.X/SCALED.RU))^2 + ((y0 -
X_init_Earth(2))*(SCALED.Y/SCALED.RU))^2 + ((z0 -
X_init_Earth(3))*(SCALED.Z/SCALED.RU))^2);
eventConditions(2) = 1e5*(((vx0 -
X_init_Earth(4))*(SCALED.Vx/CONST.MaxVel))^2 + ((vy0 -
X_init_Earth(5))*(SCALED.Vy/CONST.MaxVel))^2 + ((vz0 -
X_init_Earth(6))*(SCALED.Vz/CONST.MaxVel))^2);
eventConditions(3) = m0 - CONST.m0/SCALED.MU;

ast_used = ast_ephem(906:910,:);
```

```
data =[];
data = CartesianAst (ast_used,Ef,tf);
%posvel = CartesianPV
(tf,ast_used(:,2),ast_used(:,3),ast_used(:,4),ast_used(:,6),ast_used(:,
5),ast_used(:,7),ast_used(:,8));

%[Delta_Vf,ind] = min(((vxf - data(:,4))*(SCALED.Vx/CONST.MaxVel)).^2 +
((vyf - data(:,5))*(SCALED.Vy/CONST.MaxVel)).^2 + ((vzf -
data(:,6))*(SCALED.Vz/CONST.MaxVel)).^2);
[Delta_V,ind] = min(data(:,7)-X_init_Earth(7));

eventConditions(4) = 1e-8*(((xf - data(ind,1))*SCALED.X)^2 + ((yf-
data(ind,2))*SCALED.Y)^2 + ((zf - data(ind,3))*SCALED.Z)^2);

eventConditions(5) = 1e4*(((vxf - data(ind,4))*SCALED.Vx)^2 + ((vyf -
data(ind,5))*SCALED.Vy)^2 + ((vzf - data(ind,6))*SCALED.Vz)^2);

eventConditions(6) = M_Earth - E0 + EARTH.e * sin(E0);

eventConditions(7) = data(ind,8) - Ef + data(ind,9) * sin(Ef);

%eventConditions(8) = (t_f - t_0)/m_f;   %Gradual Cost Constraint

AstrNo=ind;
% **************** END of FILE *********************************
```

Path Function

```
function PathFunction = SPPath(primal)

% Path File
% By Dr. Pooya Sekhavat
% Last updated May 31 2007 by David Koeppel

global SCALED
global CONST

thrust_x = primal.controls(1,:);%.*SCALED.Tx;      % Thrust x direction
thrust_y = primal.controls(2,:);%.*SCALED.Ty;      % Thrust y direction
thrust_z = primal.controls(3,:);%.*SCALED.Tz;      % Thrust z direction
thrust_mag = primal.controls(4,:);%.*SCALED.TT;    % Thrust magnitude
r_mag = primal.controls(5,:);%.*SCALED.RU;

x_sp = primal.states(1,:);%.*SCALED.X;
y_sp = primal.states(2,:);%.*SCALED.Y;
z_sp = primal.states(3,:);%.*SCALED.Z;

PathFunction (1,:) = (thrust_x.*(SCALED.Tx/SCALED.TT)).^2 +
(thrust_y.*(SCALED.Ty/SCALED.TT)).^2 +
(thrust_z.*(SCALED.Tz/SCALED.TT)).^2 - thrust_mag.^2;
```

```
PathFunction (2,:) = (x_sp.*(SCALED.X/SCALED.RU)).^2
+(y_sp.*(SCALED.Y/SCALED.RU)).^2 + (z_sp.*(SCALED.Z/SCALED.RU)).^2 -
r_mag.^2;
```

Main File

```
% Spacecraft Initial Conditions in xyz

% Main File.
% By Dr. Pooya Sekhavat
% Last updated May 31 2007 by David Koeppel

clear all;

warning off MATLAB:fzero:UndeterminedSyntax

global CONST
global EARTH
global ast_ephem
global data
global SCALED
global AstrNo

CONST.AU = 1.49597870691e+08; % km      %Astronomical Unit
CONST.mu =  1.32712440018e11; % km3/s2  %Gravitational parameter of the
Sun
CONST.g = 9.80665e-3; %km/s2            %Standard acceleration due to
gravity
CONST.isp = 4000;     % s               %specific impulse
CONST.vinf = 3.5; % km/s                %hyperbolic excess velocity
CONST.DayToSec =  86400; % s            %One day in seconds
CONST.MinLaunchJD = 57023*CONST.DayToSec;   % s             %Problem
start date  00:00 01 January 2015, (MJD)
CONST.MaxArrivalJD = 64693*CONST.DayToSec;  % s             %Problem
end date    24:00 31 December 2035, (MJD)
CONST.year = 365.25 * CONST.DayToSec;       % s             %One year
in seconds
CONST.DtoR = pi/180;                        %multiplicative conversion
from Degrees to Radians
CONST.DisTol = 1000; % km
CONST.vTol = 1e-3;   % km/s
CONST.m0 = 1500;     % kg                           %initial mass 1.5 ton
= 1500 kg
CONST.mProp = 1000;  % kg                           %prop mass 1 ton=1000
kg
CONST.Tmax = 0.1e-3; % kg.km/s2                     %max thrust
ton.AU/Day^2= 0.1(N=kg.m/s^2)

% ----------------------------------------------------------------
SCALED.RU = 10*CONST.AU ; % km  = 15*AU (since a's are given in AU
units) %
SCALED.X =  10*CONST.AU ;
SCALED.Y =  10*CONST.AU ;
```

```
SCALED.Z =  10*CONST.AU ;

SCALED.Ti = 4*sqrt (SCALED.RU^3/CONST.mu);

SCALED.Vx = 4*sqrt(CONST.mu/SCALED.RU);
SCALED.Vy = 4*sqrt(CONST.mu/SCALED.RU);
SCALED.Vz = 4*sqrt(CONST.mu/SCALED.RU);

SCALED.MU = 1000;

SCALED.Tx = 1*(CONST.mu*SCALED.MU)/(SCALED.RU^2);
SCALED.Ty = 1*(CONST.mu*SCALED.MU)/(SCALED.RU^2);
SCALED.Tz = 1*(CONST.mu*SCALED.MU)/(SCALED.RU^2);
SCALED.TT = 1*(CONST.mu*SCALED.MU)/(SCALED.RU^2);
%1*(SCALED.MU*CONST.isp*CONST.g)/(SCALED.Ti);
% ----------------------------------------------------------------

EARTH.a = 0.999988049532578*CONST.AU;             % (km)
EARTH.e = 1.671681163160e-02;                     %Eccentricity
EARTH.inc = 0.8854353079654e-03 * CONST.DtoR;     %Inclination in
RADIANS!!!
EARTH.lan = 175.40647696473 * CONST.DtoR;         %Longitude of
Ascending Node in Radians!!!
EARTH.w = 287.61577546182 * CONST.DtoR;           %arg of perihelion in
RADIANS!!!
EARTH.M0 = 257.60683707535* CONST.DtoR;           %Mean Anomaly at
epoch in RADIANS!!!
EARTH.t0 = 54000*CONST.DayToSec;                  %Earth epoch time
(MJD)
CONST.MaxPos = 15*CONST.AU; %5.909335501968675e+008; %15*CONST.AU;  %
km
CONST.MinPos = 0.1*CONST.AU;
CONST.MaxVel = 100; % km/s
Xmin =
CartesianPV(CONST.MinLaunchJD,EARTH.a,EARTH.e,EARTH.inc,EARTH.w,EARTH.l
an,EARTH.M0,EARTH.t0);
% % Xmax=
CartesianPV(CONST.MaxArrivalJD,EARTH.a,EARTH.e,EARTH.inc,EARTH.w,EARTH.
lan,EARTH.M0,EARTH.t0);

%-------------------------
% Define the problem:
%-------------------------
JPL.cost           = 'SPCost';
JPL.dynamics       = 'SPDynamics';
JPL.events         = 'SPEvents';
JPL.path           = 'SPPath';

%----------------------------------------
% Setup the knots and some more constants
%----------------------------------------

t0min   = (CONST.MinLaunchJD-CONST.MinLaunchJD)/SCALED.Ti;
t0min_up    = 8*CONST.year/SCALED.Ti;
```

```
tfMax    = (CONST.MaxArrivalJD-CONST.MinLaunchJD)/SCALED.Ti;
tfMax_low    = 8*CONST.year/SCALED.Ti;
tfGuess = tfMax_low+(tfMax-tfMax_low)/2;

knots.locations      = [t0min tfGuess];      % time free problem
knots.definitions    = {'hard', 'hard'};
knots.bounds.lower   = [t0min tfMax_low];                  % [t0min
tfMax_low];
knots.bounds.upper   = [t0min_up tfMax];     % upper bound was tf0 =
fixed.
knots.numNodes       = [15];

%---------------------------------------
% Bounds
%---------------------------------------

bounds.lower.states = [-CONST.MaxPos/SCALED.X; -CONST.MaxPos/SCALED.Y;
-CONST.MaxPos/SCALED.Z; (CONST.m0-CONST.mProp)/SCALED.MU; -
CONST.MaxVel/SCALED.Vx; -CONST.MaxVel/SCALED.Vy; -
CONST.MaxVel/SCALED.Vz];
bounds.upper.states = [ CONST.MaxPos/SCALED.X;  CONST.MaxPos/SCALED.Y;
CONST.MaxPos/SCALED.Z; CONST.m0/SCALED.MU;  CONST.MaxVel/SCALED.Vx;
CONST.MaxVel/SCALED.Vy;  CONST.MaxVel/SCALED.Vz];

bounds.lower.controls = [-CONST.Tmax/SCALED.Tx; -CONST.Tmax/SCALED.Ty;
-CONST.Tmax/SCALED.Tz;  0; CONST.MinPos/SCALED.RU];
bounds.upper.controls = [CONST.Tmax/SCALED.Tx; CONST.Tmax/SCALED.Ty;
CONST.Tmax/SCALED.Tz; CONST.Tmax/SCALED.TT; CONST.MaxPos/SCALED.RU];

bounds.lower.events = [0; 0; 0; 0; 0; 0; 0];
bounds.upper.events = [0; 1e5*CONST.vinf^2/CONST.MaxVel^2; 0; 1e-
8*CONST.DisTol^2; 1e4*CONST.vTol^2; 0; 0; 0.09];

bounds.lower.path = [0; 0];      % convexifying the nonconvex cone;
lower bound = - inf
bounds.upper.path = [0; 0];

bounds.lower.parameters = [0; 0];
bounds.upper.parameters = [40*2*pi; 50*2*pi];

%----------------------------
% Provide a guess
%----------------------------

guess.states(1,:)   = [Xmin(1)/SCALED.X, CONST.MaxPos/SCALED.X];
guess.states(2,:)   = [Xmin(2)/SCALED.Y, CONST.MaxPos/SCALED.Y];
guess.states(3,:)   = [Xmin(3)/SCALED.Z, CONST.MaxPos/SCALED.Z];
guess.states(4,:)   = [CONST.m0/SCALED.MU, (CONST.m0-
CONST.mProp)/SCALED.MU];
guess.states(5,:)   = [Xmin(4)/SCALED.Vx, CONST.MaxVel/SCALED.Vx];
guess.states(6,:)   = [Xmin(5)/SCALED.Vy, CONST.MaxVel/SCALED.Vy];
guess.states(7,:)   = [Xmin(6)/SCALED.Vz, CONST.MaxVel/SCALED.Vz];
```

```
guess.controls(1,:) = [0, CONST.Tmax/SCALED.Tx];
guess.controls(2,:) = [0, CONST.Tmax/SCALED.Ty];
guess.controls(3,:) = [0, CONST.Tmax/SCALED.Tz];
guess.controls(4,:) = [0,
norm([CONST.Tmax;CONST.Tmax;CONST.Tmax],2)/SCALED.TT];
guess.controls(5,:) = [norm([Xmin(1);Xmin(2);Xmin(3)],2)/SCALED.RU,
norm([CONST.MaxPos;CONST.MaxPos;CONST.MaxPos],2)/SCALED.RU];

guess.time      = [t0min tfGuess];

guess.parameters (1) = [15*2*pi];
guess.parameters (2) = [20*2*pi];

beginCPU = cputime;
[cost, primal, dual] = dido(JPL, knots, bounds, guess);
RunTime = (cputime - beginCPU)/60

cost

Feasibility (primal);

% Bootstrap
%
% guess.states = primal.states;
% guess.controls = primal.controls;
% guess.time = primal.nodes;
% knots.locations = [primal.nodes(1), primal.nodes(end)];
% knots.numNodes    = [60];
% % --------------------------
% beginCPU = cputime;
% [cost, primal, dual] = dido(JPL, knots, bounds, guess);
% RunTime = (cputime - beginCPU)/60
%
% cost
%
% Feasibility (primal);
```

LIST OF REFERENCES

Boden, Daryl G. "Introduction to Astrodynamics," in *Spacecraft Mission Analysis and Design,* edited by Wiley J Larson and James R. Wertz. El Segundo, CA: Microcosm Press, 1999.

Bryson, Arthur E. *Dynamic Optimization.* Menlo Park, CA: Addison-Wesley Longman, Inc., 1999.

Gong, Qi, Ross, I.M., Kang, Wei and, Fahroo, F. "On the Pseudospectral Covector Mapping Theorem for Nonlinear Optimal Control." Paper presented at the 45th IEEE Conference on Decision & Control, San Diego, CA, Dec. 13-15, 2006.

Petropoulos, Anastassios E. "Problem Description for the 2nd Global Trajectory Optimization Competition," Outer Planets Mission Analysis Group, Jet Propulsion Laboratory, 6 November 2006.

Ross, I. M. *Control and Optimization: An Introduction to Priciples and Applications, Electronic Eddition.* Naval Postgraduate School, Monterey, CA, December 2005.

Ross, I. M. and Fahroo, F. "Legendre Pseudospectral Approximations of Optimal Control Problems." Lecture Notes in Control and Information Sciences (2003).

Ross, I. M. and Fahroo, F. "Pseudospectral Methods for Optimal Motion Planning of Differentially Flat Systems." Paper presented at IEEE Conference on Transactions on Automatic Control, Las Vegas, NV, Dec. 2002.

Ross, I. M. and Fahroo, F. "User's Manual for DIDO 2001: A MATLAB Application Package for Dynamic Optimization" Tec. Rep. NPS Technical Report AA-01-003, Department of Aeronautics and Astronautics. Monterey, CA: Naval Postgraduate School, 2001.

Sellers, Jerry Jon, *Understanding Space: An Introduction to Astronautics.* Boston: McGraw Hill Custom Publishing, 2004.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California